© 2024 Heting Gao

UNSUPERVISED SPEECH TECHNOLOGY FOR LOW-RESOURCE LANGUAGES

BY

HETING GAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Professor Mark Hasegawa-Johnson, Chair
Professor Paris Smaragdis
Professor Suma Bhat
Professor Yan Tang

ABSTRACT

Deep neural network based speech processing systems have found widespread applications in daily life, being employed for tasks such as automatic speech recognition (ASR), text-to-speech (TTS) synthesis, spoken language understanding (SLU), *etc.* With a sufficient amount of parallel speech-text training data, these systems attain performance levels comparable to, or in some cases, even better than human capabilities. However, such sufficient data assumption holds for only resource-rich languages such as English and Mandarin Chinese, and is unrealistic for many existing low-resource languages, posing a challenge for these systems to attain similar high performance. It is therefore meaningful to improve speech processing systems in such conditions to make speech technology accessible to a broader population.

Unsupervised learning has been an active research field to mitigate data sparsity of low-resource languages. Depending on different source-target scenarios, unsupervised learning can be classified into four categories: (1) self-supervised learning (SSL), (2) modality matching, (3) unsupervised transfer learning, and (4) unsupervised multimodal learning. This thesis introduces six projects that leverage unsupervised learning methods to improve speech processing systems.

The first project pretrains the SSL models on monolingual, cross-lingual, and multimodal data to study the cross-lingual transferability of SSL models. The second project improves the SSL representations using synthetic speech generated by a diffusion-based unit-to-speech synthesizer. The third project falls under modality matching, where we build the first unsupervised speech-to-text system using unsupervised automatic speech recognition technology. The fourth project falls under unsupervised transfer learning, where we improve zero-shot phonetic recognition system using language embeddings derived from external linguistic databases, without requiring any training data from the target languages. The fifth project also falls under transfer learning, where we build a multimodal few-shot SLU system by prompting a frozen pretrained language model with text and acoustic embeddings. The sixth project falls under unsupervised transfer learning, where we improve the current grapheme-to-phoneme (G2P) transducer by integrating the grapheme-to-phoneme model with a unit-to-phoneme (U2P) model, aiming to regularize G2P model outputs without relying on ground truth phoneme transcripts as training labels.

This thesis demonstrates that unsupervised learning methods can significantly improve the performance of speech recognition, speech synthesis, and speech understanding in low-resourced application scenarios.

To my beloved father and mother.

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Mark Hasegawa-Johnson, for his vision, insight and guidance through my Ph.D journey.

I would like to thank my mentors, Doctor Zhang Yang and Doctor Kaizhi Qian, for their guidance and help in my research.

I would like to thank my colleagues Junrui Ni, John Harvill, Mahir Moshed, Xulin Fan and Xiuwen Zheng and others in the SST group for having interesting discussions and happy memories at the Beckman 2137.

I would like to thank my father and mother, Kaike Gao and Zhijun Luo, who raise me and educate me, and my family, who love me and support me.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

As of now, there are 7,168 languages in the world according to Ethnologue [1], among which only 4,178 languages have a developed writing system and the remaining 2,990 languages are likely spoken only. In this context, speech can be regarded as one of the most enduring and prevalent forms of communication throughout human history. Speech conveys ideas, intentions, emotions, and more. To extract these contents from speech and, conversely, to synthesize speech from these contents, speech processing systems have been meticulously designed and developed.

1.1 Neural Speech Systems for Low-Resource Languages

Modern end-to-end neural network based speech processing systems can generally be divided into two components: the front-end and the back-end.

The front-end handles data preprocessing tasks, including speech denoising, speech enhancement, text normalization, grapheme-to-phoneme conversion, *etc.* In particular, the grapheme-to-phoneme (G2P) transducer, though it does not directly deal with speech, is an important component of speech systems. This is because many speech systems are trained using phoneme transcripts rather than raw text (grapheme transcripts) and it is commonly reported that phoneme transcripts typically result in a lower error rate than using text transcripts [2, 3].

The back-end handles actual speech processing, such as automatic speech recognition (ASR) systems [4, 5, 6, 7, 8, 9, 10, 11, 12], text-to-speech (TTS) systems [13, 14, 15, 16, 17, 18, 19], spoken language understanding (SLU) systems [20, 21, 22, 23, 24, 25, 26], *etc.* These systems have achieved great success in transcribing speech into text, generating human-like speech from text, and determining the speaker's intention.

While attaining performance levels comparable to or in some cases, even better than human capabilities for resource-rich languages such as English and Mandarin, where large parallel speech-text corpora are available, these systems do not work as well for low-resource languages, a category encompassing the majority of the world's languages. Out of the 7,186 languages in the world, only 492 languages are considered institutional, meaning they are used and sustained by institutions beyond the home and community. The four most spoken languages - English, Mandarin Chinese, Hindi, and Spanish - account for a combined population of approximately 3.7 billion speakers. This represents roughly only half of the world's population. CommonVoice [27], the largest public speech-text corpus collected by crowdsourcing, initially contained around 2,500 hours of speech across 29 languages when released in 2019. As of now, it has expanded to include approximately 30,000 hours from 107 languages. It is worth noting, however, that 89 of these included languages have speech data of less than 100 hours. The abundance of low-resource languages, coupled with their data scarcity, presents a serious challenge for neural networks to achieve comparable accuracy levels to those achieved in resource-rich languages. Speech-processing systems on low-resource languages have, therefore, become an active research field, which aims to improve existing systems under limited training data conditions and to make speech technology accessible to a wider variety of users who speak a minority language.

1.2 Unsupervised Speech Technology

Unsupervised learning methods have been actively studied to address the data sparsity of low-resource languages. Unlike supervised learning, these methods alleviate the need for human efforts in labeling data.

Denote the input source of the neural network as X and the output target as Y. Depending on different source-target scenarios, unsupervised learning can be classified into four categories: (1) self-supervised learning (SSL), (2) modality matching, (3) unsupervised transfer learning, and (4) unsupervised multimodal learning.

1.2.1 Self-Supervised Learning

When the learning task involves only the source X itself rather than relying on the target Y labeled by humans, unsupervised learning falls under self-supervised learning. The goal of the self-supervised learner is to learn the source data distribution $\mathbb{P}(X)$. The precision with which it needs to learn $\mathbb{P}(X)$ depends on the downstream application: some applications need high-precision models of the temporal dynamics of quantized modes of $\mathbb{P}(X)$ [11], others need high-precision models of locally linearized approximations [28], *etc.* In order to focus the learner on a type of learning relevant to the downstream task, many recent classes of algorithms learn $\mathbb{P}(\mathcal{T}(X)|X)$, where $\mathcal{T}(X)$ captures some intuition about the aspects of $\mathbb{P}(X)$ that are most important to the downstream application. In speech processing, Wav2vec2 [11] and HuBERT [12] are probably the two most successful examples of self-supervised learning methods. The training objective of Wav2vec2 is to predict the quantized codeword of the current frame features given the surrounding frame features and the training objective of HuBERT is to predict the K-Means cluster centroids of the current frame features given the surrounding features. This thesis includes one project that studies the performance of SSL model trained using monolingual, cross-lingual, and multilingual data and one project that improves the SSL pretraining of Wav2vec2 and HuBERT models using synthetic data generated by diffusion models.

1.2.2 Modality Matching

When both source X and target Y are present but no parallel X-Y data is available for training, unsupervised learning to learn the conditional distribution $\mathbb{P}(Y|X)$ falls under modality matching of X and Y. Wang *et al.*, 2023 [29] prove conditions under which $\mathbb{P}(Y|X)$ can be learned from unpaired examples of discrete X and discrete Y, but to the best of our knowledge, nobody has yet proven conditions under which $\mathbb{P}(Y|X)$ can be learned from unpaired data for continuous X and/or continuous Y. Despite the lack of theoretical support for this task, unsupervised modality matching has received extensive experimental study in recent years, with examples including unsupervised machine translation [30, 31, 32], unsupervised speech recognition [33, 34, 35, 36, 37, 2, 29] *etc.* This thesis includes one project that builds the first unsupervised speech synthesis system by leveraging unsupervised ASR that matches the modality between the text and the speech.

1.2.3 Unsupervised Transfer Learning

When no label Y is available for training and we still want to learn $\mathbb{P}(Y|X)$, we can train the model in a related domain where $X^{'}$ and $Y^{'}$ are available and adapt it to the intended domain. We refer to this approach as unsupervised transfer learning. This thesis includes two projects on unsupervised transfer learning. The first project improves the zero-shot transferability of a multilingual phonetic ASR system by incorporating language embeddings extracted from external linguistic databases, without the need for any training data from the target language. The second project exploits the transferability of language models pretrained on massive text corpora to perform few-shot SLU without being explicitly trained on the SLU task.

1.2.4 Unsupervised Multimodal Learning

Another strategy to improve the model on the intended domain is to incorporate information from additional modalities. Suppose an auxiliary modality Z exhibits correlation with the target Y. Incorporating this additional modality can enhance the model's performance. We refer to this approach as unsupervised multimodal learning. This thesis includes one project on unsupervised multimodal learning that improves the prediction of a G2P transducer by incorporating additional acoustic information, without access to groundtruth phonetic transcripts during training.

1.3 Organization of This Thesis

This thesis explores using unsupervised methods to improve speech processing systems on low-resource languages. Six projects are introduced where unsupervised speech technology, including self-supervised learning, modality matching, unsupervised transfer learning, and unsupervised multimodal learning, is applied to improve ASR, TTS, and SLU systems.

The first project studies the performance of SSL models trained using monolingual, cross-lingual, and multilingual data and explores the best setting to pretrain these models for downstream ASR tasks.

The second project explores improving the Wav2vec2 and HuBERT pretraining using synthetic data generated by diffusion models. While SSL in speech has greatly reduced the reliance of speech processing systems on annotated corpora, the success of SSL still hinges on the availability of a large-scale unannotated corpus, which is still often impractical for many low-resource languages or under privacy concerns. In this project, we investigate whether existing SSL methods have been underutilizing the information in pretraining and explore ways to improve their information efficiency. Motivated by the recent success of diffusion models in capturing the abundant information in data, we propose DiffS4L, a synthetic speech SSL algorithm based on diffusion models. DiffS4L introduces a diffusion model, which learns from a given small pretraining dataset and expands it into a much larger synthetic dataset with different levels of variations. The synthetic dataset is then used to pretrain SSL models. Our experiments show that DiffS4L can significantly improve the performance of SSL models, such as reducing the WER of the HuBERT pretrained model by 6.26 percentage points in the English ASR task. Notably, even the nonsensical babbles generated by the diffusion model can account for a significant portion of the performance improvement, which indicates the strong capability of diffusion models in capturing coherent information in speech that has been overlooked by SSL methods.

The third project falls under modality matching, where we build the first unsupervised speech-to-text system using unsupervised automatic speech recognition technology. An unsupervised TTS system learns to generate the speech waveform corresponding to any written sentence in a language by observing: (1) a collection of untranscribed speech waveforms in that language; (2) a collection of texts written in that language without access to any transcribed speech. Developing such a system can significantly improve the availability of speech technology to languages without a large amount of parallel speech and text data. This paper proposes an unsupervised TTS system that trains on the pseudo-transcripts from an unsupervised ASR system. Our unsupervised system can achieve comparable performance to the supervised system in seven languages with about 10 to 20 hours of speech each. A careful study on the effect of text units and vocoders has also been conducted to better understand what factors may affect unsupervised TTS performance.

The fourth project falls under unsupervised transfer learning, where we improve zero-shot phonetic recognition system using language embeddings derived from external linguistic databases, without requiring any training data from the target languages. Many existing languages are too sparsely resourced for monolingual deep learning networks to achieve high accuracy. Multilingual phonetic recognition systems mitigate data sparsity issues by training models on data from multiple languages and learning a speech-to-phone or speech-to-text model universal to all languages. However, despite their good performance on the seen training languages, multilingual systems have poor performance on unseen languages. This project argues that in the real world, even an unseen language has metadata: linguists can tell us the language name, its language family and, usually, its phoneme inventory. Even with no transcribed speech, it is possible to train a language embedding using only data from language typologies (phylogenetic node and phoneme inventory) that reduces ASR error rates. Experiments on a 20-language corpus show that our methods achieve phonetic token error rate (PTER) reduction on all the unseen test languages. An ablation study shows that using the wrong language embedding usually harms PTER if the two languages are from different language families. However, even the wrong language embedding often improves PTER if the language embedding belongs to another member of the same language family.

The fifth project also falls under transfer learning, where we build a multimodal few-shot SLU system by prompting a frozen pretrained language model with text and acoustic embeddings. Large-scale auto-regressive language models pretrained on massive text have demonstrated their impressive ability to perform new natural language tasks with only a few text examples, without the need for finetuning. Recent studies further show that such a few-shot learning ability can be extended to the text-image setting by training an encoder to encode the images into embeddings functioning like the text embeddings of the language model. Interested in exploring the possibility of transferring the few-shot learning ability to the audio-text setting, we propose a novel speech understanding framework, WavPrompt , where we finetune a wav2vec model to generate a sequence of audio embeddings understood by the language model. We show that WavPrompt is a few-shot learner that can perform speech understanding tasks better than a naive text baseline. We conduct detailed ablation studies on different components and hyperparameters to empirically identify the best model configuration. In addition, we conduct a non-speech understanding experiment to show WavPrompt can extract more information than just the transcriptions. The sixth project falls under unsupervised transfer learning, where we improve the current grapheme-to-phoneme (G2P) transducer by integrating the grapheme-to-phoneme model with a unit-to-phoneme (U2P) model, aiming to regularize G2P model outputs without relying on ground truth phoneme transcripts as training labels. Most phoneme transcripts are generated using forced alignment: typically a G2P transducer is applied to text sequences to generate candidate phoneme transcripts, which are then time-aligned to the waveform using an acoustic model. This project demonstrates, for the first time, simultaneous optimization of the G2P, the acoustic model, and the acoustic alignment to a corpus. To this end, we propose G2PU, a joint CTC-attention model consisting of an encoder-decoder G2P network and an encoder-CTC U2P network, where the units are extracted from speech. We demonstrate that the G2P and U2P, operating in parallel, produce lower phone error rates than those of state-of-the-art open-source G2P and forced alignment systems. Furthermore, although the G2P and U2P are trained using parallel speech and text, their synergy can be generalized to text-only test corpora if we also train a grapheme-to-unit (G2U) network that generates speech units from text in the absence of parallel speech. Our G2PU model is trained using phoneme transcripts generated by a teacher G2P tool. Our experiments on Chinese and Japanese show that G2PU reduces the phoneme error rate by 7% to 29% relative to its teacher. Finally, we include case studies to provide insights into the system's workings.

The remainder of this thesis is organized as follows. Chapter 2 introduces the background of speech processing systems and reviews methods commonly employed to address data sparsity issues. Chapter 3 introduces a project on comparing the performance of pretraining SSL models using monolingual, cross-lingual and multilingual data. Chapter 4 introduces a project on SSL, where the monolingual pretraining of acoustic SSL models is improved with synthetic data generated using a diffusion model. Chapter 5 introduces a project on modality matching, where the first unsupervised TTS system is built to match text and speech modalities by leveraging unsupervised ASR technology. Chapter 6 introduces a project on unsupervised transfer learning, where linguistic knowledge extracted from language databases is utilized to improve the performance of a multilingual phonetic ASR system in zero-shot cross-lingual phonetic recognition. Chapter 7 introduces another project on unsupervised transfer learning, where an SLU system is constructed using a frozen language model that is prompted with continuous speech prompts generated by a finetuned Wav2vec2. Chapter 8 introduces a project that leverages both text and speech modalities to improve existing G2P transducers, a crucial component of phoneme-based speech systems. Chapter 9 discusses the pros and cons of these proposed methods. Finally, Chapter 10 summarizes this thesis.

CHAPTER 2

BACKGROUND

2.1 Speech Processing Systems

This section introduces related works on automatic speech recognition (ASR) systems, text-to-speech (TTS) synthesis systems, and spoken language understanding (SLU) systems.

2.1.1 Automatic Speech Recognition Systems

Traditional hidden Markov model (HMM) based ASR systems [38] decompose the ASR task into four steps: acoustic feature extraction from speech waveform, HMM-based acoustic modeling, count-based language modeling and decoding [39]. As computing power has advanced, neural networks have gradually replaced HMM networks in acoustic modeling due to their capacity to model complex patterns and scalability to large datasets. Successful neural ASR systems include connectionist temporal classification (CTC) [4] based DeepSpeech [6] and attention-based Listen-Attend-Spell (LAS) [7]. Recently, self-supervised learning (SSL) has brought another advance in ASR, especially for low-resource languages where the amount of parallel speech-text data is limited. Instead of being trained directly on the parallel data in an end-to-end fashion, the ASR model is first pretrained on large unlabeled speech corpora to learn better speech representations and is then finetuned on the limited speech-text data. Examples of SSL models include Wav2vec2 [11], HuBERT [12] and XLSR [40]. Building on the advancements in large language models and prompt engineering, Whisper [41] trained on massive speech corpora achieves state-of-the-art performance on speech recognition in many languages.

2.1.2 Text-to-Speech Systems

Modern deep neural network based TTS systems are usually split into two components: a text encoder to predict speech features (typically spectrogram) from text and a vocoder to predict waveforms from speech features. The text encoder usually contains a duration predictor to align the text token sequence with the speech features as the former is usually much shorter than the latter. WaveNet [13] is a generative audio model that uses dilated causal convolution blocks to directly model the temporal dependency of raw audio waveforms and generate speech samples in an autoregressive manner. WaveNet usually serves as the vocoder and can also serve as the text encoder if equipped with an external duration predictor. Due to the autoregressive nature of WaveNet, the generation cost of WaveNet is prohibitively high. Other non-autoregressive speech synthesis methods such as flow-based WaveGlow [42] and GAN-based Parallel-WaveGAN [43] and HiFi-GAN [44] have been proposed to reduce the generation cost while maintaining a satisfactory generation quality. Recently, diffusion models [45, 46] has also been employed for speech synthesis such as Grad-TTS [47] and ProDiff [48]

2.1.3 Spoken Language Understanding Systems

SLU is similar to natural language understanding (NLU), differing primarily in that it involves speech input instead of text input. In a broader sense, SLU and NLU include a range of sub-tasks, such as intent classification, slot filling, named entity recognition, emotion classification, sentient classification, translation, question answering (QA), *etc.* The most common approach to SLU is a pipelined approach in which a speech-to-text model is followed by an NLU model. This approach is common because the accuracy benefits gained by integrating the speech-to-text and the text understanding modules into a single end-to-end training pipeline are usually small, but despite being small, those benefits have been repeatedly demonstrated [49, 20, 21, 22, 23]. Additional textual information can be incorporated into the SLU model to improve performance. WCN-BERT [20] inputs the word confusion lattice to the BERT language model to provide richer information about possible alternate transcriptions of the speech signal instead of only the best hypothesis.

Sari *et al.*, 2020 [49] leverages non-parallel speech and text data to improve SLU performance by introducing a shared speech-text intent classifier. Sunder *et al.*, 2022 [23] pretrains the SLU model with additional text information to teach the speech encoder to produce semantic-aware speech embeddings for better SLU performance.

Recently, with the paradigm of natural language processing shift towards large language models (LLM) [50, 51, 52], spoken language understanding is unified to the text generation task using language models [53, 54, 55, 56]; instead of prediction of the one-hot intent labels, LLM-based SLU models treat the intent label as text and directly predict the sequence of subword tokens that constitutes the intent word or phrase.

Another change brought by LLM is the generalizability to out-of-domain problems. Due to the limited amount of parallel speech-text data, previous SLU methods use small models (typically around a few million parameters) and can only perform in-domain tasks, where models are trained and tested on the same task. For example, PENGI [53] model uses a pretrained GPT2 [51] with 124 million parameters and is trained and tested on the same audio caption datasets. LMV [57] model uses a pretrained PALM [58] with 350 million parameters and is trained and tested on the same spoken QA datasets. It seems small language models are sufficient for in-domain generation. Leveraging the prior distribution learned from the extensive unlabeled training text, LLMs broaden the objective of SLU from in-domain generation to out-of-domain generation, where models are trained and tested on different tasks. For example, The LTU [55] and X-LLM [56] models perform out-of-domain QA about audio and speech respectively, and the LLM involved typically contains six to seven billion parameters, which is more than 20 times larger than the language model used LMV and PENGI.

2.1.4 Grapheme-to-Phoneme Systems

Many works on G2P have focused on transcription at the word level and are evaluated using a lexicon with ground-truth phonetic transcriptions. LSTM-G2P [59] is the first to approach word-level G2P using a long short-term memory network (LSTM) based sequence-to-sequence (S2S) architecture. More recently, LSTM-based G2Ps were replaced with convolutional neural networks based G2P [60] and Transformer-based G2Ps [61] for improved performance on word-level G2P tasks. ByT5 [62] compares SentencePiece [63] and byte-level tokenization schemes when finetuning pretrained large language models for word-level G2P.

Recently, research efforts have shifted towards sentence-level G2P due to the ability to incorporate contextual information. T5G2P [64] finetunes a text-to-text transfer transformer (T5) model to perform sentence-level G2P conversion and found improved performance over an encoder-decoder baseline. Wang, 2021 [65] uses special annotations of both correct and incorrect pronunciations (and parts of speech) in training data to improve disambiguation of polyphones – or Chinese characters with multiple pronunciations – in sentences when using a bidirectional LSTM-based text encoder. Rezaei *et al.*, 2022 [66] use a bidirectional GRU-based encoder-decoder module that yields lower WER in Persian compared to a fully recurrent or fully transformer-based network. SoundChoice [67] uses an S2S architecture for sentence-level G2P that takes advantage of contextual word embeddings and a homograph loss.

2.2 Speech Technology for Low-Resource Languages

This section introduces related works on different methods that have been applied to low-resource languages to mitigate the data sparsity issue.

2.2.1 Self-Supervised Learning

Self-supervised large pretrained language models have advanced the state-of-the-art (SOTA) performance in various natural language processing [68, 69] and ASR [11, 12, 70] tasks. For example, large language models such as BERT [68], RoBERTa [69] and GPT [51, 52] trained on large unlabeled text corpora capture complex linguistic relationships and meaning between words and generate better contextual representations for words, phrases, and sentences that improve the performance in sentiment analysis, paraphrasing, and question answering, *etc* [71].

This strategy is also shown to be effective in the speech domain as well. Speech models such as Wav2vec2 [11] and HuBERT [12] are trained on unlabeled speech waveforms using contrastive predictive coding or masked prediction of hidden units. These models extract high-quality contextual speech features from raw audio signals that improve a variety of speech-related downstream tasks such as phoneme recognition, emotion classification, speaker identification, *etc* [72]. Besides improving state-of-the-art performance, pretrained language models greatly reduce the requirement for labeled data. Finetuned on only 10 hours of parallel speech-text data, the Wav2vec2 trained on 960-hour LibriSpeech speech dataset [73] can achieve a word error rate (WER) of 10.9% on the `dev-clean` set and the WER can be further reduced to 3.8% when Wav2vec2 is decoded with a 4-gram language model. Without pretraining, previous ASR systems such as DeepSpeech2 require 2400 hours of transcribed English speech to achieve similar WER on English [6].

2.2.2 Modality Matching

Training good end-to-end speech-to-text or text-to-speech models usually requires a large amount of paired data. However, such data is expensive to obtain and is usually limited for most of the languages in the world. Modality matching seeks to learn the conditional distribution of the target labels given the source data using training examples drawn from the marginal source distribution and from the marginal target distribution, but without any paired examples drawn from the joint distribution of the source and target. In machine translation, this is achieved by matching the marginal distribution of text tokens in the source language and the target language [30, 31]. In speech recognition, this is achieved by matching the marginal distribution of the text tokens and discretized speech tokens as in unsupervised ASR work [35, 37]. Similar methods can be applied to unsupervised speech-to-sign-language systems [74].

2.2.3 Transfer Learning

In the transfer learning framework, the model is trained on one or a few tasks and is then applied or finetuned on the target task. The general problem of transfer learning is to figure out what type of knowledge learned from one distribution can be transferred to another distribution. In this sense, self-supervised learning can be considered as a special case of transfer learning: self-supervised models transfer the knowledge learned with self-supervised criteria to the target task with supervised criteria. The methods to extract transferable knowledge can be partially or fully fixing the model based on prior knowledge, learning the transfer functions between the pretraining and the target distributions, or some combination of both.

In the speech recognition domain, transfer learning is widely employed in multilingual ASR systems where the ASR models are trained on a range of languages and then applied to the target language with or without finetuning. Since a large number of languages do not have enough parallel speech and text data, deep learning models trained on them often produce high error rates [75]. Multilingual speech recognition mitigates the data sparsity by training the network on a combined dataset from several languages. The network usually has a common encoder that extracts acoustic information from audio features and can either have a common decoder with a shared phoneme inventory [76, 77] or language-specific decoders with private phone [78, 79, 80] or character inventories [81, 82, 83].

Language or dialect embedding has been shown to improve the knowledge transfer for multilingual ASR systems [84, 85, 86, 87]. The embedding can be a one-hot vector specifying language ID [84, 86] or a vector learned from acoustic data under a standard multilingual model [85, 87] and can be used as additional input features to the network [84, 86], as adapter modules for language-specific adjustments [86] or as interpolation weights for the encoder [85].

2.2.4 Multimodal Learning

The concept of multimodal learning is inspired by human perception, which involves senses such as sight, sound, touch, and smell, each of which is considered one modality. Common modalities studied in machine learning include speech, text, vision, *etc.*

Combining speech and visual modalities, VG-HuBERT [88] train a HuBERT or Wav2vec2 model to associate speech with images using paired image and spoken caption data. It is reported that the learned speech representations have a good performance in the SUPERB benchmark and can be clustered to retrieve good word segmentations. SpeechCLIP finetunes the pretrained HuBERT and CLIP model to align paired images and spoken captions and achieves SOTA on image-speech retrieval and zero-shot speech-text retrieval [89]. Combining the information in the different modalities, multimodal learning generally yields models with better robustness compared to single-modal learning [90].

Combining speech and text modalities, SpeechT5 [91] trains a unified-modal representation and achieves SOTA performance on a variety of downstream speech processing tasks. SeamlessM4T [92] trains a semi-supervised multilingual sequence-to-sequence model for a variety speech-to-text and text-to-speech tasks.

Combining vision and text modalities, BERT-gen [93] train a vision encoder to embed the images into feature vectors understood by a pretrained BERT model to perform visual question generation tasks. Tsimpoukelli *et al.*, 2021 [94] propose Frozen, where a large frozen auto-regressive language model is prompted to perform zero-shot and few-shot visual question-answering tasks via visual prompts. Different from the aforementioned multimodal methods, Frozen is not directly trained on the intended task and hence fits into the category of unsupervised multimodal learning.

CHAPTER 3

COMPARING MONO-, CROSS- AND MULTI-LINGUAL PRETRAINING OF SELF-SUPERVISED SPEECH MODELS

In order to study the self-supervised speech models on different languages, we[1] pretrain six self-supervised Wav2vec2 [11] models on speech corpora of six different languages, namely, English (EN), Bulgarian (BG), Chinese (ZH), Russian (RU), Arabic (AR), and Japanese (JP), and compare their performances in monolingual, cross-lingual, and multilingual settings. In the monolingual setting, the Wav2vec2 models are pretrained on the target language directly and finetuned on the same language. In the cross-lingual setting, the Wav2vec2 model is pretrained on one language, English in this work, and is finetuned on the target language. In the multilingual setting, the Wav2vec2 models are pretrained on data from a set of different languages and finetuned on the target language. In this work, we use the officially released XLSR model [40] which is pretrained on 56k hours of raw speech data from 53 languages.

3.1 Experiments Setup

We obtain raw speech data for pretraining Wav2vec2 models from LibriSpeech [73] for English; VoxPopuli [95] for Bulgarian; United Nations Proceedings Speech [96] for Chinese, Russian and Arabic; and LaboroTVSpeech [97] for Japanese. We obtain transcribed speech data from LibriSpeech for English; GlobalPhone [98] for Bulgarian and Chinese; GALE Broadcast News Datasets [99] for Arabic; Corpus of Spontaneous Japanese [100] for Japanese; and Russian LibriSpeech [101] for Russian.

We uniformly sample 960 hours of raw speech from pretraining corpora for

[1] The experiments described in this chapter were conducted as part of an unpublished small-group project, with co-authors Mahir Morshed, Shuju Shi, Liming Wang, and Junkai Wu. The code is available in https://github.com/Hertin/Wav2vec

Wav2vec2 pretraining and sample 10 hours of transcribed speech for finetuning. We additionally sample 5 hours of transcribed speech for validation and another 5 hours for testing. Such pretraining and finetuning data split follows that of the Wav2vec2 work [11] for a fair comparison to the official English Wav2vec2 model. The models are pretrained and fine-tuned using four 16 GB NVIDIA V100 GPU. To emulate the 64-GPU pretraining of official Wav2vec2, we set the update frequency to 16. The maximum token per batch is reduced from 1.4 million to 1 million due to the 16 GB GPU memory limit.

3.2 Results

The main results are shown in Table 3.1. The Wav2vec2 models are evaluated in character error rate (CER) rather than word error rate (WER) because the computation of word error rate involves additional complexity dependent on the writing system, which varies across languages and is unavailable for some languages such as Chinese and Japanese, where the division of text into words is not well defined. The experiments are denoted in the format of "{pretrain language}-{finetuning language}".

3.2.1 Monolingual vs. Cross-lingual vs. Multilingual Pretraining

By comparing the monolingual experiments to cross-lingual and multilingual experiments, we find that the monolingual Wav2vec2 pretrained on 960 hours of target language beats the cross-lingual Wav2vec2 pretrained on the same amount of data and even the multilingual XLSR model pretrained on 56k hours of data. The exception of xlsr-en can be explained by the fact that the multilingual dataset contains about 44k hours of English speech, which is much more than 960 hours in the monolingual setting. On the other side, although not as good as monolingual training, cross-lingual pretraining and multilingual pretraining do provide decent-performing Wav2vec2 models that have error rates close to the monolingual pretraining. This finding indicates the existence of an imperfect common acoustic model that can achieve decent performance when monolingual data are not enough.

Table 3.1: Comparison of CER and WER of mono-lingual, cross-lingual and multi-lingual Wav2vec2 pretraining-finetuning

Mono	Dev CER	Dev WER	Test CER	Test PER
en-en	3.51	9.87	2.97	3.19
bg-bg	1.85	8.78	1.89	11.28
zh-zh	10.43	-	10.60	15.03
ru-ru	5.57	23.06	6.98	7.87
ar-ar	3.62	12.32	3.45	4.20
jp-jp	10.38	-	9.91	3.32
Cross				
en-en	3.51	9.87	2.97	3.20
en-bg	3.48	17.68	3.47	14.86
en-zh	15.20	-	15.41	18.05
en-ru	5.57	27.92	5.59	6.10
en-ar	6.49	20.41	5.47	6.55
en-jp	16.08	-	16.33	4.98
Multi				
xlsr-en	1.91	6.58	1.91	2.29
xlsr-bg	2.67	13.79	2.70	17.38
xlsr-zh	14.15	-	14.56	16.10
xlsr-ru	5.84	28.21	4.84	5.26
xlsr-ar	4.67	17.56	4.58	5.24
xlsr-jp	14.67	-	14.35	4.51

3.2.2 Grapheme vs. Phoneme Transcripts

The CER of Chinese Wav2vec2 models (zh-zh, en-zh and xlsr-zh), and Japanese models (jp-jp, en-jp and xlsr-jp) are obviously higher than others. This can be attributed to the writing systems used by Chinese and Japanese. Chinese and Japanese writing systems contain Chinese characters and Japanese Kanji, which greatly increases the character inventory size to the magnitude of thousands while the character inventory sizes of other languages are in the magnitude of tens. This results in a much larger output space. To verify this hypothesis, we use LanguageNet grapheme-to-phoneme transducer [102] to convert the grapheme transcripts to international phonetic alphabet (IPA) transcripts and use the IPA transcripts to

Table 3.2: Comparison of PER of monolingual, cross-lingual and multilingual Wav2vec2 finetuning on different grapheme or phoneme transcripts.

	ZH-ZH	EN-ZH	XLSR-ZH
Char	10.60	15.41	14.56
IPA	15.03	18.05	16.10
IPA w/o tone	14.56	16.94	15.44
Pinyin	**2.96**	**3.80**	**4.02**
Pinyin w/o tone	**2.31**	**2.97**	**2.79**

finetune the Wav2vec2 models. The resulting models are evaluated using phoneme error rates (PER) shown in Test PER column in Table 3.1. We observe the PERs of Japanese models are lowered but the PERs of Chinese models are even higher.

3.2.3 Case Study on Chinese Transcripts

We hypothesize that the increased PERs of Chinese models are due to the quality of the IPA transcripts produced by the Chinese G2P tools. Therefore, we conduct a case study comparing the performance of the rule-based FST LanguageNet and the neural network based G2PW [103] on Mandarin Chinese. We finetune the pretrained Wav2vec2 models using the Chinese-character-based grapheme transcripts (Char), the IPA transcripts generated by LanguageNet and pinyin transcripts by G2PW. For IPA transcripts and pinyin transcripts, we experiment with and without tone annotations. The results are shown in Table 3.2. By comparing the error rates of Char and pinyin, we observe that the phoneme-based transcripts indeed have lower error rates than the grapheme-based transcripts when used to finetune Wav2vec2 models. The IPA annotation and the pinyin annotation are equivalent; one can be converted to the other. Theoretically, the error rate using IPA transcripts should be similar to that using pinyin transcripts. However, we observe a large performance gap between using IPA and using pinyin. These findings suggest the phoneme transcript can result in a lower error rate compared to the text, as it better matches the actual pronunciation in the speech, and that the quality of the phoneme transcript produced by the G2P tools is critical to the performance of the fine-tuned Wav2vec2 models.

3.3 Summary

This study compares the SSL models trained on monolingual, cross-lingual, and multilingual data. The experimental results suggest that monolingual pretraining achieves the best downstream ASR performance. On the other side, cross-lingual pretraining achieves decent performance on the test languages, making it the preferred option when the amount of speech data available for the target language is not sufficient.

The case study on Chinese suggests the quality of the G2P tool is essential to the performance of the finetuned SSL models.

CHAPTER 4

SPEECH SELF-SUPERVISED LEARNING USING DIFFUSION MODEL SYNTHETIC DATA

4.1 Introduction

Self-supervised learning (SSL) in speech has greatly reduced the reliance of speech processing systems on large-scale annotated corpora. By pretraining a speech representation network on a large-scale unannotated dataset, SSL models only require a relatively small annotated dataset for finetuning, which has significantly improved the efficiency and feasibility of speech processing, particularly for low-resource languages. However, the success of such methods still hinges on the availability of a large-scale unannotated corpus. For example, the training of HuBERT [12], one of the most widely-used speech pretraining models, typically requires that the unannotated corpus contains at least 1,000 hours of speech. If the dataset size drops to 100 hours, it tends to produce degenerate results. Yet, in many scenarios, obtaining such a large-scale dataset is still impractical due to various constraints, *e.g.*, low-resource languages, privacy concerns, *etc.*

Such limitations have prompted us to re-examine SSL from an *information efficiency* perspective. Essentially, if we[1] consider the pretraining dataset as a source of information about speech data at various levels (from phonetics to semantics), then SSL can be seen as a way to extract that information. In situations where the pretraining dataset is limited, it becomes crucial to maximize the amount of information captured from the dataset to achieve the best performance in downstream tasks. This raises the question – do existing SSL techniques have a high enough information efficiency? Could there be additional information that SSL models fail to capture, which would

[1] The project described in this chapter is part of a manuscript currently under review, with co-authors Kaizhi Qian, Junrui Ni, Chuang Gan, Mark A. Hasegawa-Johnson, Shiyu Chang, and Yang Zhang. The code is available in `https://anonymous.4open.science/status/DiffS4L-CE41`

otherwise contribute to a better performance in downstream tasks?

On the other hand, generative models are also often considered models that capture the distributional information about data. Recently, diffusion models [104, 105], with their superior performance in computer vision, have quickly attracted wide research attention. Researchers have found that compared to other generative models, diffusion models can generate samples with much better global coherence [106] and local details [107], an indication that diffusion models may be able to capture more complete information from a limited dataset that could complement those learnable by existing SSL methods.

Motivated by this, in this paper, we conduct an extensive exploration of using synthetic data generated by diffusion models to improve the performance of existing SSL methods in a low-resource setting. In particular, we propose a Synthetic Speech Self-Supervised Learning algorithm called DiffS4L. DiffS4L introduces a diffusion model, which learns from a given small pretraining dataset and then expands it into a much larger synthetic dataset. The new dataset contains synthetic speech utterances with different levels of variations, ranging from identical utterances in the original small dataset to near-complete babbles. Finally, the synthetic dataset is used to pretrain SSL models using existing algorithms. Since the diffusion model only has access to the information in the original real dataset, the entire process can be viewed as restructuring and recreating the information in the original pretraining dataset into a more digestible form for existing SSL methods.

Our experiments on DiffS4L reveal many interesting findings. With only 100 hours of real data, DiffS4L can significantly improve the performance of existing SSL algorithms over models pretrained on the real data alone. In English ASR, for example, DiffS4L can reduce the WER by 6.26 percentage points for HuBERT pretrained models, which is a 26.4% relative improvement. Notably, the babbles generated by diffusion models, which are complete nonsense to humans, can account for a significant portion of the performance improvement, while babbles generated by other generative models, such as WaveNet [13], only deteriorate the performance. These findings suggest the information in pretraining datasets has been underutilized, and diffusion models are very effective in capturing the information that has been overlooked by existing SSL training methods and other generative models.

4.2 Related Works

4.2.1 Data Augmentation with Synthetic Data

Training neural networks with synthetic data to improve performance has been extensively studied in various computer vision tasks such as visual representation learning [108, 109, 110, 111], image classification [112, 113], object detection [114, 115, 116], anomaly detection [117], semantic segmentation [118, 119], action recognition [120, 121], visual reasoning [122], and embodied perception [123, 124, 125]. Recently, this direction has also been studied in NLP tasks such as machine translation [126] and language model pretraining [127] and finetuning [128].

Augmenting datasets with synthetic data has been shown effective in improving speech processing systems. One research direction modifies speech waveforms by adding random noise [129], warping spectrogram, masking blocks of spectrograms in frequency and time domains [9], modifying pitch and adding reverberation [130], and disentangling speaker information from speech content [70].

Another line of research augments the dataset using speech data generated from speech synthesizers and reports improvement on speech translations [131], fake audio detection [132], speech recognition [133, 134, 135, 136, 137, 138, 139, 140], *etc.* Zheng *et al.*, 2021 [141] use synthetic data to improve the recognition of out-of-vocabulary words in ASR systems. Zhao *et al.*, 2022 [131] generate synthetic training data by retrieving and stitching clips from a spoken vocabulary bank. Li *et al.*, 2018 [135] train a Tacotron-2 [15] conditioned on Global Style Tokens [142] to generate speech with different speaking styles. Jin *et al.*, 2022 [138] use a GAN-based generator conditioned on dysarthric speech characteristics to generate synthetic speech for dysarthric ASR. Krug *et al.*, 2022 [139] generate articulatory speech for phoneme recognition. These works improve traditional task-specific speech systems by generating additional paired speech and text data while our work aims to improve general-purpose self-supervised speech representations without additional text data that benefits downstream ASR and other speech-related tasks.

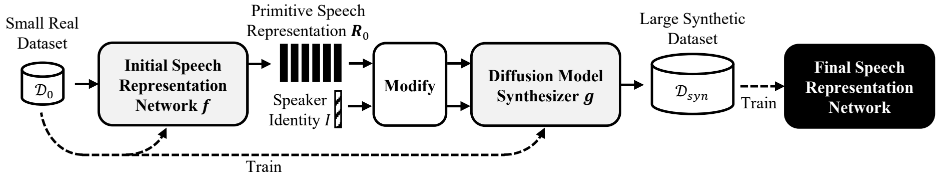

Small Real Dataset
$\mathcal{D}_0$
Initial Speech Representation Network f
Primitive Speech Representation R_0
Speaker Identity I
Modify
Diffusion Model Synthesizer g
Large Synthetic Dataset
$\mathcal{D}_{syn}$
Train
Final Speech Representation Network
Train

Figure 4.1: The algorithm overview. Solid arrows represent the data flow that generates the synthetic dataset. Dashed arrows mark the dataset on which each network is trained.

4.2.2 Denoising Diffusion Probabilistic Models for Speech

Denoising diffusion probabilistic models (DDPMs) have recently demonstrated great power in image synthesis [104, 107] and image impainting [143] tasks. Recently various DDPM-based vocoders and text-to-speech (TTS) synthesizers have been proposed [144, 145, 146, 147, 148, 48] and achieved high quality. WaveGrad [144] and DiffWave [146] are two concurrent works that study the DDPM-based vocoder to synthesize audio waveform from spectrograms; WaveGrad uses a neural architecture inspired by GAN-TTS [149] and DiffWave inspired by WaveNet. FastDiff [148] and Prodiff [48] are end-to-end TTS systems that use FastSpeech [17], a transformer-based TTS encoder, to extract text feature to condition the DDPM and adopts the noise scheduling algorithm proposed in BBDM [147] to shorten the sampling steps for fast speech synthesis.

4.3 The DiffS4L Algorithm

In this section, we will formally introduce our proposed DiffS4L algorithm to improve the speech self-supervised learning problem under a low-resource scenario.

4.3.1 Problem Formulation

Denote a speech utterance as $\boldsymbol{X}$. Speech self-supervised learning involves training a speech representation network on an unannotated dataset, $\mathcal{D}_0$. A successful speech self-supervised learning scheme would typically require that $\mathcal{D}_0$ contains as much as around 1,000 hours of speech or more. In this paper, we will focus on the case where $\mathcal{D}_0$ contains much fewer data, *e.g.* only around 100 hours of speech.

4.3.2 The Algorithm Overview

DiffS4L approaches the problem by synthesizing a much larger dataset $\mathcal{D}_{syn}$, which is then used to pre-train the speech representation network. As shown in Figure 4.1, the algorithm consists of four steps.

Step 1: Use $\mathcal{D}_0$ to train an initial speech representation network $\boldsymbol{f}_0(\cdot)$, which can produce a primitive speech representation, denoted as $\boldsymbol{R}_0 = \boldsymbol{f}_0(\boldsymbol{X})$.

Step 2: Use $\mathcal{D}_0$ to train a diffusion-model-based speech synthesizer $\boldsymbol{g}(\cdot)$, which generates speech $\boldsymbol{X}$ conditional on the partially masked primitive speech representation $\boldsymbol{R}_0$ and speaker identity, denoted as I, *i.e.*, $\tilde{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{R}_0, I)$.

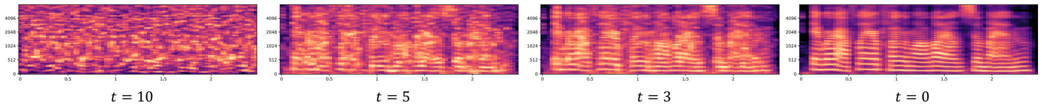
Step 3: For each utterance $\boldsymbol{X}$ in $\mathcal{D}_0$, manipulate its speech representation $\boldsymbol{R}_0$ and speaker identity I, and then fed to the speech synthesizer to generate utterances with different levels of variations. Denote the resulting dataset as D_{syn}.

Step 4: Use D_{syn} to train a new speech representation network.

It is worth noting that the diffusion model only has access to the original pretraining dataset $\mathcal{D}_0$ during training and generation, so the synthetic dataset $\mathcal{D}_{syn}$ would contain no more information than $\mathcal{D}_0$, but may restructure and recreate it in a way that is more beneficial for SSL with existing methods. The following subsection will provide more details on steps 1-3, respectively.

4.3.3 Primitive Speech Representation

In our setting, the size of $\mathcal{D}_0$ is very small, and many existing SSL methods tend to degenerate in such a low-resource scenario. In contrast, Wav2vec [11] can still produce non-degenerate results, so we adopt the Wav2vec for our primitive speech representation learning. Note that the algorithm used to train the final speech representation network (step 4) need not be the same as the one for the primitive speech representation learning. After the Wav2vec is trained, we elicit the 5th-layer feature and quantize it into 500



$t = 10$ $t = 5$ $t = 3$ $t = 0$

Figure 4.2: The intermediate denoising spectrograms of a 20-step DDPM denoising process. As t decreases to zero, the spectrograms transform from white noise to a speech spectrogram.

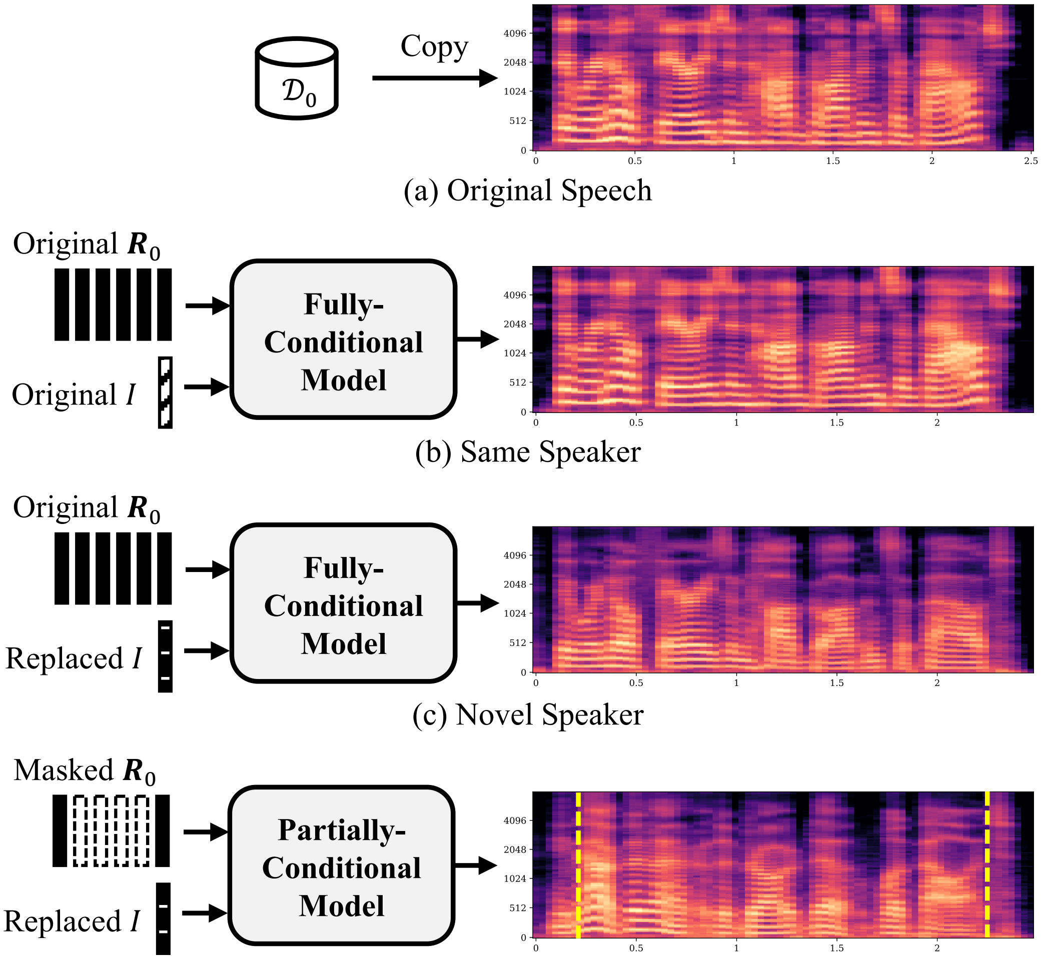
classes using k-means, which becomes the primitive speech representation $\boldsymbol{R}_0$ for the subsequent steps.

4.3.4 Diffusion-Model-Based Speech Synthesizer

Diffusion models refer to a family of generative models that denoise from noise signals into clean signals through multiple denoising steps. In this work, we adopt the canonical denoising diffusion probabilistic model (DDPM) [104] to generate a speech spectrogram. Specifically, DDPM introduces a set of intermediate variables, denoted as $\boldsymbol{X}_{0:T}$, where $\boldsymbol{X}_0$ is the original speech spectrogram. Each $\boldsymbol{X}_t$ is generated by downscaling and adding Gaussian noise to $\boldsymbol{X}_{t-1}$, so that by the time it reaches $\boldsymbol{X}_T$, the signal becomes very similar to Gaussian white noise.

Speech spectrograms are generated by the backward denoising process, which introduces a denoising neural network that learns to predict $\boldsymbol{X}_{t-1}$ from $\boldsymbol{X}_t$. Therefore, by feeding a Gaussian white noise $\boldsymbol{X}_T$ to the denoising network and going through all the T denoising steps, clean speech spectrograms can be generated, as visualized in Figure 4.2. We encourage the readers to refer to the original papers for further details and derivations.

In this work, we introduce two diffusion models, a *fully-conditional model* and a *partially-conditional model*. For the fully-conditional model, the denoising network is conditioned upon the entire primitive speech representation $\boldsymbol{R}_0$, so that the diffusion model will generate speech that follows the content depicted in $\boldsymbol{R}_0$. For the partially-conditional model, the denoising network is still conditioned upon $\boldsymbol{R}_0$, but with a consecutive span of 80% of the frames masked out. In this case, the diffusion model will follow the content in $\boldsymbol{R}_0$ only where it is unmasked, and try to generate novel content that fits into the given context at the remaining frames. These two models are both crucial in generating synthetic data with different levels of variations.



(a) Original Speech

(b) Same Speaker

(c) Novel Speaker

(d) Novel Content

Figure 4.3: An example of synthetic utterances at different levels of variations. The transcription of the original utterance is ‘There were no ferries and hobgoblins about’. The yellow dashed lines on the spectrogram in (d) mark the boundaries of the masks on $\boldsymbol{R}_0$.

Besides $\boldsymbol{R}_0$, both models are also conditional on speaker labels. Speaker labels can be either one-hot vectors or speaker embeddings produced by a pre-trained speaker embedding network, depending on whether D_0 comes with speaker labels. We will compare different conditioning settings in Section 4.4.

Since the diffusion model generates spectrograms, a vocoder is needed to convert the spectrograms into speech waveforms. To this end, we adopt a HiFiGAN [44], which is also trained only on the small dataset $\mathcal{D}_0$.

4.3.5 Synthetic Speech Generation

The synthetic speech generation uses the original speech dataset $\mathcal{D}_0$ as seeds. Specifically, we first draw a speech utterance from $\mathcal{D}_0$ as the seed speech, elicit its primitive speech representation $\boldsymbol{R}_0$ and speaker identity I, and then generate a synthetic utterance by feeding a modified version of these conditioning variables to the diffusion model synthesizer. The primary consideration of designing the modification schemes for the conditioning variables is the tradeoff between novelty and naturalness – if the generated speech is identical to the original utterance, we can achieve maximum naturalness but introduce no new information to the dataset; if the generated speech is a complete babble, we can introduce maximum novelty but may significantly compromise naturalness. Therefore, we introduce the following four different levels of novelty, as shown in Figure 4.3:

- **Original Speech (O):** The seed speech is directly copied to the synthetic dataset without modification. No resynthesis is involved for this level.
- **Same Speaker (SS):** $\boldsymbol{R}_0$ and I are fed as is to the fully-conditional diffusion model. The resulting synthetic speech is almost the same as the seed speech. However, since $\boldsymbol{R}_0$ tends to obscure the pitch information, the synthetic speech will be in a different intonation, as shown in Figure 4.3(a).
- **Novel Speaker (NS):** $\boldsymbol{R}_0$ is still fed as is to the fully-conditional diffusion model, but I is replaced with a different speaker ID. As a result, the synthetic would still have the same content, but in a different voice and intonation, as shown in Figure 4.3(c).
- **Novel Content (NC):** We mask out a consecutive span of 80% frames in $\boldsymbol{R}_0$ and replace I before feeding them to the partially-conditional diffusion model. As shown in Figure 4.3, the synthetic speech is almost completely different from the seed speech in terms of content, speaker, and prosody, except for the content information in the 20% unmasked frames. The utterances are almost nonsensical babbles to human listeners. We are thus interested in seeing whether utterances at this high level of randomness could still contribute to SSL.

As we will show, all four levels of the speech are beneficial for the subsequent speech pretraining and thus should all be included into $\mathcal{D}_{syn}$ with appropriate ratios.

Table 4.1: Main results on (a) English automatic speech recognition and (b) SUPERB benchmark. The bolded results show the best performance among all but the topline models.

	English ASR		SUPERB								
			KS	IC	SID	ER	QbE	SF		ASV	SD
Task/Metric	CER↓	WER↓	ACC↑	ACC↑	ACC↑	ACC↑	MTWV↑	F1↑	CER↓	EER↓	DER↓
High-Resource Setting (960-hour real speech)											
Wav2vec2-Real	3.18	10.49	96.23	92.35	66.20	60.55	0.0233	87.64	25.37	6.67	6.65
HuBERT-Real	3.03	10.30	96.30	98.26	**66.27**	60.74	0.0736	88.53	25.20	5.80	6.30
Wav2vec2-DiffS4L	2.98	9.93	96.17	94.73	65.79	61.29	0.0630	88.50	24.71	6.60	6.63
HuBERT-DiffS4L	**2.95**	**9.87**	**96.47**	**98.50**	64.36	**61.40**	**0.0766**	**88.93**	**24.03**	**5.78**	**6.26**
Low-Resource Setting (100-hour real speech)											
Wav2vec2-Real	7.37	23.48	91.92	88.64	47.68	58.99	0.0311	81.31	37.06	8.78	8.45
HuBERT-Real	7.43	23.71	91.82	78.43	**57.53**	61.84	0.0419	78.87	40.69	8.91	8.53
Wav2vec2-Aug	6.92	22.06	92.18	92.83	48.65	58.34	0.0377	81.99	36.39	8.37	8.84
Wav2vec2-DiffS4L	**5.19**	16.67	93.57	91.01	45.41	59.86	0.0331	83.13	33.60	8.02	7.14
Wav2vec2-OneHot	**5.19**	**16.65**	93.23	91.41	48.94	61.64	0.0364	83.00	34.64	8.14	7.28
HuBERT-DiffS4L	5.33	17.45	**94.68**	**95.94**	44.22	62.02	**0.0469**	**84.61**	32.68	**7.42**	**7.09**
HuBERT-OneHot	5.36	17.47	94.26	95.89	44.25	**62.60**	0.0445	83.98	**32.33**	7.64	7.44

4.4 Experiments

In this section, we will present our experimental results on training different SSL models integrating DiffS4L.

4.4.1 Implementation Details

The entire training pipeline is constructed based on two existing code repositories: Fairseq[2] [150], ProDiff[3] [48].

Pretraining SSL models We use Fairseq [150] to pretrain all the speech SSL models. In particular, We use the same hyperparameter as specified in the `wav2vec2_base_librispeech` and `hubert_base_librispeech` configuration files in Fairseq to pretrain Wav2vec2 and HuBERT respectively. The training of Wav2vec2 models requires 64 Tesla V100-SXM2-32GB GPUs and that of HuBERT models requires 32 GPUs. The pretraining dataset for both models is the 100-hour seed dataset, $\mathcal{D}_0$ for the initial speech representation network and is the augmented dataset $\mathcal{D}_{syn}$ for the final speech representation network as described in Sec 4.4.7 Dataset Composition. The

[2] `https://github.com/facebookresearch/fairseq`

[3] `https://github.com/Rongjiehuang/ProDiff`

Table 4.2: ASR results (CER/WER) on selected languages from MLS and CommonVoice. The languages are (from left to right, top to bottom) English, German, Spanish, French, Italian, Dutch, Polish, Portuguese, Bashki, Central Kurdish, Welsh, Meadow Mari, Swahili, and Tamil.

Languages	EN	DE	ES	FR	IT
Wav2vec-100R	7.4/23.5	8.3/30.4	7.1/27.2	16.2/45.5	8.3/35.1
Wav2vec-DiffS4L	**5.2/16.8**	**6.4/23.3**	**4.5/16.7**	**11.9/34.8**	**6.2/27.2**
Languages	NL	PL	PO	BA	CKB
Wav2vec-100R	17.8/50.9	11.4/44.2	13.8/45.8	10.2/43.8	7.2/39.0
Wav2vec-DiffS4L	**14.7/44.8**	**7.1/31.0**	**8.9/37.1**	**8.9/37.1**	**6.7/29.7**
Languages	CY	MHR	SW	TA	
Wav2vec-100R	20.6/62.1	10.7/45.4	8.8/31.5	9.2/47.2	
Wav2vec-DiffS4L	**16.7/52.3**	**9.4/37.5**	**7.0/25.9**	**7.5/41.0**	

SSL models are trained for 400k updates with a learning rate of 5×10^{-4}. Each batch contains 1.4M audio samples. The checkpoint with the best validation loss is selected for downstream tasks.

Finetuning SSL models We use the `base_10h` configuration in Fairseq for Wav2vec2 and HuBERT fine-tuning on a 10-hour limited supervision dataset. We follow the same finetuning procedure as in [11] and [12] where we add a linear projection layer on top and finetune with the CTC loss. The model is trained for 40k updates on two V100-SXM2-32GB GPUs with each batch containing 3.2M audio samples and a learning rate of 5×10^{-5}. The checkpoint with the best CER on the validation set is selected for further evaluation.

Training Diffusion Speech Synthesizer The speech synthesizer consists of a Fastspeech2 encoder and a 20-step DDPM model. The fastspeech2 encoder contains 4 Transformer encoder layers each with 4 heads. Using the initial speech representation network, we extracted the speech units from $\mathcal{D}_0$ and substitute them for the text inputs. We replace the Duration Predictor with an upsampling network consisting of a transposed convolution with a kernel size of 9, a stride of 5, and a padding of 2, followed by a convolution layer with a kernel size of 8, a stride of 5, and a padding of 2, that resamples the HuBERT units from 50Hz to 62.5Hz to match the length of 80-bin mel-spectrogram. The Fastspeech2 encoder encodes the speech units into hidden embeddings, which are combined with the broadcasted speaker embeddings to condition the training and inference of the DDPM model. The speech synthesizer is trained for 200k iterations using one V100-SXM2-32GB GPU with a batch size of 64 and a learning rate of 1. The synthesizer is optimized for the weighted sum of $\mathcal{L}_1$ reconstruction loss and structural similarity index (SSIM) loss [48] with the weight being 0.5 for each loss. We use adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ and inverse square root scheduler with 2000 warmup updates.

We use a HifiGAN vocoder[4] [44] to convert mel-spectrogram to waveform. The vocoder is trained on the same seed dataset $\mathcal{D}_0$ for 1M iteration using four V100-SXM2-32GB GPU.

4.4.2 Configurations

The majority of our experiments and ablation studies are performed in English, where the seed dataset $\mathcal{D}_0$ is the `train-clean-100` subset of the LibriSpeech-960 dataset [73]. The synthetic dataset consists of 960 hours of speech containing: 1) the 100 hours of real speech in $\mathcal{D}_0$; 2) 430 hours of SS/NS speech, which is generated by replacing the speaker ID with a uniformly randomly chosen one from all the speakers in $\mathcal{D}_0$ (can be the same as the original speaker); and 3) 430 hours of NC speech. We will evaluate the effect of changing the composition of the dataset in Section 4.4.7. Experiments in other languages are discussed in Section 4.4.4.

We use $\mathcal{D}_{syn}$ to pretrain two models, Wav2vec2 and HuBERT. The training configuration for Wav2vec2 is the same as in the initial speech representation learning, except for the dataset. For HuBERT, we adopt two rounds of training, following the 500-cluster configurations described in [12].

For comparison, we introduce two **topline models**, denoted as Wav2vec2-960R and HuBERT-960R, which are Wav2vec2 and HuBERT trained on *960-hours of real data*. Since the topline systems have access to much more information, we expect they can achieve better performance. We also intro-

[4] `https://github.com/jik876/hifi-gan`

duce two **baseline models**, denoted as Wav2vec2-100R and HuBERT-100R, which are Wav2vec2 and HuBERT trained on $\mathcal{D}_0$ only. We are interested to see how far DiffS4L models are away from the topline and baseline models.

4.4.3 Automatic Speech Recognition (ASR)

To see whether DiffS4L can extract additional information to improve the pretraining performance, we first finetune all the pretraining models to the English ASR task on 10 hours of annotated data from LibriLight [151]. The ASR performance is evaluated by two metrics, character error rate (CER) and word error rate (WER). Note that we do not use language models in all the ASR results. The results are shown in Table 4.1 section (a). As shown, not only can DiffS4L improve the performance, but it can also improve it by a large margin. In particular, DiffS4L is able to reduce the CER by over 2 percentage points and WER by over 6 percentage points for both Wave2vec and HuBERT models compared to the corresponding baselines. Notably, DiffS4L can bring both metrics down to the midpoint between the topline and baseline results, which shows the great potential that even 100 hours of pretraining data can have compared to what has been realized.

4.4.4 Extension to Other Languages

To test whether the performance improvement of DiffS4L can generalize to other languages, we select all the seven non-English languages from the Mulingual LibriSpeech (MLS) dataset [152] and six languages from the CommonVoice dataset [27]. The languages in the CommonVoice dataset are chosen based on the criterion that they have just over 100 hours of validated data in the dataset. For each language in MLS, we sample 100 hours from training split for pretraining and use the limited supervision subset for finetuning. Both cases use the provided dev and test split for validation and testing. For each language in CommonVoice, we create a 100-hour split for pretraining, and a 10-hour split for finetuning. The provided dev and test split are used for validation and testing, respectively. We only evaluate the Wav2vec2 systems due to the substantial time cost for pretraining and due to our observation that the relative improvements in both Wav2vec2 and HuBERT are similar. Also, since most of these languages do not have 960 hours of data in the dataset, we cannot compute the topline results, so we show only the baseline and DiffS4L models.

Table 4.2 demonstrates a consistent performance advantage of DiffS4L across all the languages. In particular, DiffS4L can reduce the CER by an average of 2.6 percentage points, and WER by an average of 8.3 percentage points, which is a significant improvement for ASR. Notice that these languages are from different language families and each has very unique phonetic, lexical, and syntactic structures, so these results show that the diffusion models can successfully capture various structures in all these languages.

4.4.5 Extension to Other Tasks

So far, our evaluations have only focused on the ASR tasks. We would also like to see whether the performance improvement of DiffS4L can be generalized to other tasks. To this end, we introduce the SUPERB benchmark [72], which is a collection of speech-processing tasks. We evaluate our models on KS (keyword spotting), IC (intent classification), SID (speaker identification), ER (emotion recognition), Qbe (query by example spoken term detection), SF (slot filling), ASV (automatic speaker verification) and SD (speaker diarization), We did not include the ASR and PR (phoneme recognition), as we have evaluated on extensive similar tasks in the previous sections. The results are shown in Table 4.1 section (b). Note that the topline results are copied from the original paper [72].

The results consistently demonstrate that DiffS4L-based models outperform the baseline models in almost all the tasks, affirming the versatility of DiffS4L. Interestingly, DiffS4L underperforms the baselines in the speaker identification (SID) task. One potential explanation is that the speaker replacement operation when generating the DS and NC data may obscure the speaker information. Between the two models, HuBERT has a slight performance advantage, consistent with the observations in [72].

Table 4.3: ASR Performance on improving multilingual XLSR-53 and XLSR-128 models.

Model	ba	ckb
XLSR-53	6.98/32.54	5.29/26.99
XLSR-53-100R	6.94/31.91	5.05/26.41
XLSR-53-DiffS4L	**6.61/30.11**	**4.71/24.55**
XLSR-128	6.69/31.28	4.62/24.46
XLSR-128-100R	6.45/30.28	4.59/25.00
XLSR-128-DiffS4L	**6.32/29.77**	**4.29/21.69**

4.4.6 Extension to Large Multi-lingual Pretraining

So far, all our experiments are performed on models pre-trained in at most 960 hours of English only. We would like to find out whether DiffS4L is still useful if the pre-trained model sees even more data in many languages. To this end, we select two multilingual pre-trained models, XLSR-53 [40] and XLSR-128 [153], which were pretrained 53 and 128 languages respectively. We then select two low-resource languages, Bashki and Central Kurdish. For each language and each pre-trained model, we derive two other pre-trained models, one by further pre-training the XLSR models on 100 hours of the low-resource data, and the other by further pre-training on 100 hours of low-resource data plus the DiffS4L-augmented data. All three pre-trained models are then finetuned on the ASR task with 10 hours of labeled data, and the results are reported in Table 4.3, which shows a clear advantage of DiffS4L despite the abundance of pre-training data.

4.4.7 Dataset Composition

In all the previous experiments, we fixed the synthetic dataset composition to 100 hours of real speech, 430 hours of SS/NS speech, and 430 hours of NC speech. In the following, we will use $x + y + z$ to denote x hours of real speech, y hours of SS/NS, and z hours of NC. So the aforementioned dataset composition can be abbreviated as 100+430+430.

To better understand the contribution of each component, we perform an ablation study where we change the dataset composition. Since the time cost for training HuBERT systems, which need to be trained for two rounds, is much larger than that of Wav2vec2, and there are a large number of different settings, to keep our computation tractable, we will only perform experiments on Wave2vec2.0 and on the English ASR tasks in all the remaining ablation studies.

In our first experiment, we fix the total hours of the dataset to 960 and fix hours of real data to 100, but we vary the ratio of the SS/NS and NC from 100+860+0 to 100+0+860. The results are shown in Figure 4.4. There are two important observations. First, the performance curve exhibits a U-shape, with the lowest CER and WER achieved when both SS/NS and NC are of comparable amounts. This indicates that both the recombination of speaker information and the innovation of content plays crucial roles in improving the performance of SSL models. In particular, note that NC data is essentially nonsensical babbles reflecting the limited knowledge of phone transitions learned by the diffusion models from the small real dataset, and that one of the purposes of SSL models is also to learn the phone transition structures. The fact that the nonsensical babble can still help the SSL performance implies the DDPM synthesizer is able to generate, in a manner consistent with the original dataset, examples of phone transitions and speech sound variants that are not well represented in the original dataset.

Our second observation of Figure 4.4 is that comparing the two extreme cases, the performance without the SS/NS data (the left endpoint) is worse than that without the NC data (the right endpoint). Recall that SS/NS data are generated conditional on the true content information and therefore are of high quality, whereas NC data generally sound messier and noisier. This observation may be ascribed to the quality differences in the synthetic data.

Now that we have verified the contribution of synthesizing novel content, we will investigate the effect of synthesizing novel speaker combinations in the next experiment. In particular, we start with the standard dataset composition, *i.e.* 100+430+430, but instead, we do not replace the speaker in any of the synthesis types; hence there is no longer NS data and the NC data are of reduced speaker variations. The corresponding result, shown in Table 4.5 (Wav2vec-SS), shows a marked performance degradation (1.7 percentage points in CER and 5.0 percentage points in WER) compared to the standard dataset composition, which verifies that the novel speaker combination is crucial to the performance.

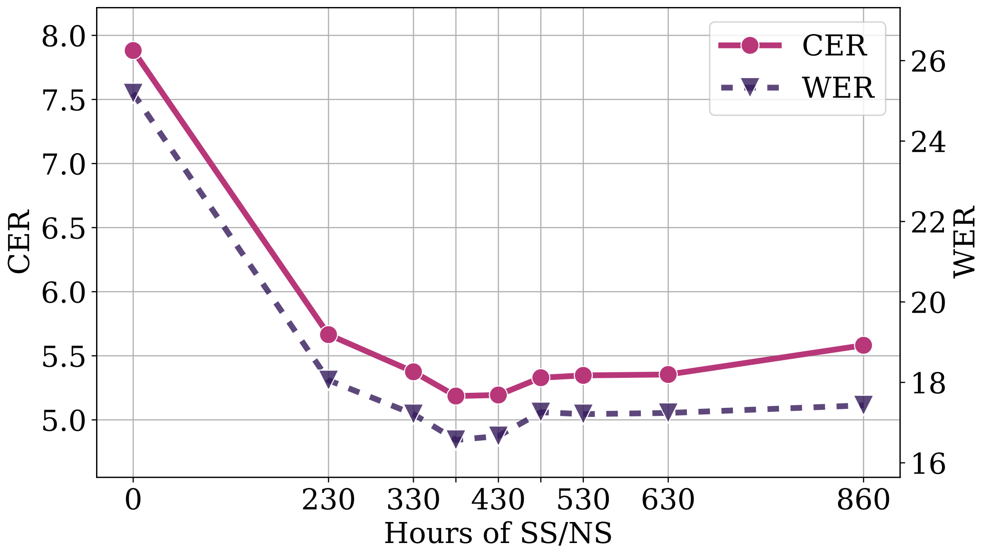


Figure 4.4: Performance over different dataset compositions by varying the ratios of SS/NS and NC speech while fixing the amount of real speech, ranging from 100+860+0 to 100+0+860.

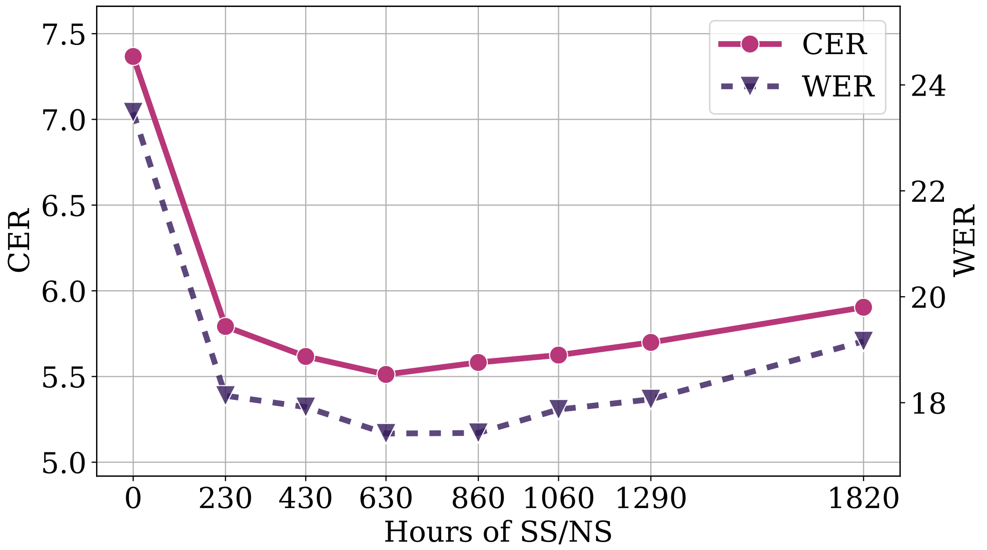


Figure 4.5: Performance over different synthetic dataset sizes, 100+x+0, where x ranges from 0 to 1820.

Finally, to test the contribution of including the original dataset, we remove the real data and expand the synthetic data proportionally to 960 hours, *i.e.* 0+480+480. The result, as shown in Table 4.5 (Wav2vec-NoReal), shows

Table 4.4: English ASR performance in CER/WER of Wav2vec2 pretrained on DiffS4L-generated data versus that on Wavenet-generated data.

Model	100+860+0	100+430+430	100+0+860
DiffS4L	5.58/17.43	5.19/16.67	7.88/24.91
Wavenet	6.07/18.95	6.71/21.90	12.57/39.02

an even larger performance degradation. In fact, we find that without the real data, the SSL training has difficulty converging. This shows that including the real data is essential for successful SSL training with synthetic data.

4.4.8 Dataset Size

Since we have verified that synthetic data improve SSL training, a natural follow-up question is whether increasing the amount of synthetic data is always beneficial. To answer this question, we fix the real data to 100 hours and NC data to 0 hours, but vary the hours of SS/NS data, *i.e.*, 100+x+0, with x ranging from 0 to 1820. Figure 4.5 shows the corresponding Wav2vec2 results on English ASR. As can be observed, the performance does not always improve as the amount of synthetic data increases. When the amount of synthetic data is small, increasing synthetic data can drastically improve performance. However, as synthetic data continues to increase, the performance gradually saturates and then starts to degrade, with the optimal performance achieved at around 630 hours. Combining the previous results, we can conclude that although adding synthetic data can inject new knowledge and variations, adding too much can dilute the contribution of the real data, which have been shown essential for the training, and, hence, will negatively impact the performance.

4.4.9 Comparison with WaveNet

To test whether the diffusion model can be replaced with other generative models, we conduct a new set of experiments by switching from the diffusion model to WaveNet [13]. We also trained two versions of WaveNet, a

Table 4.5: Additional ablation studies on the English ASR task (CER/WER), including removing speaker variations, removing real speech, and replacing the speaker embedding with one-hot.

Model	English ASR
Wav2vec2-DiffS4L	5.19/16.67
Wav2vec2-SS	6.91/21.69
Wav2vec2-NoReal	18.26/52.79

fully-conditional one on the speech representation $\boldsymbol{R}_0$ and speaker identity I, and a partially-conditional one only on I. To generate the SS+DS data, we randomly replace the I the same way as in the diffusion model experiments, and then feed the $\boldsymbol{R}_0$ and replaced I to the fully conditional WaveNet. To generate the NC babbles, we randomly select 3 seconds of real speech as the prompt and use the partially-conditional WaveNet to generate the subsequent waveforms conditional on I. We then pretrain Wav2vec2 using three synthetic data compositions, 100+860+0, 100+430+430, and 100+0+860, and compare the English ASR results with the diffusion model counterparts, as shown in Table 4.4. As shown, both WaveNet results are worse than the corresponding diffusion model ones, which suggests that WaveNet-generated speech may have a lower overall quality. More importantly, unlike the case of diffusion models, where an adequate amount of NC babble improves performance, WaveNet-generated NC babbles are always detrimental to performance, and the more NC babble is introduced, the worse the performance. This comparison underlines the unique advantage of the diffusion model in generating babble that better captures the inherent structure in speech, which is essential to the success of DiffS4L, as already shown in Figure 4.5.

4.4.10 Speaker Identity

To test the impact of different forms of the speaker identity I, we retrain a diffusion model with I replaced with the one-hot speaker embedding, rather than the embedding generated from the pretrained speaker embedding network. We reevaluate the English ASR performance on the 100+430+430 dataset composition. As can be observed from Table 4.1 (row Wav2vec2-OneHot and HuBERT-OneHot), there is no significant difference between the performances of different forms of the speaker identity, which rules out the possibility that the superior performance of the DiffS4L is due to the additional knowledge leaked in from the speaker embedding network.

4.4.11 Full ASR results on LibriSpeech English

We provide the full ASR results on LibriSpeech English dataset in Table 4.6, including the CER/WER evaluated on dev-clean, dev-other, test-clean and test-other subset of LibriSpeech-960 dataset. The experiments are labeled as ‘EN-X-Y’, where ‘X’ denotes the number of hours of untranscribed real speech for pretraining and ‘Y’ denotes the number of hours of transcribed real speech for finetuning. We use the 10-hour limited supervision set from LibriLight for Y=10 and the ‘train-clean-100’ subset from LibriSpeech for Y=100. We additionally provide the results of Wav2vec-Aug in EN-100-10 for comparison with Wav2vec2 and Wav2vec-DiffS4L.

Language Models It has been widely known that introducing language models will rectify the ASR results, and thus tends to obscure the performance gap between different ASR algorithms. We, therefore, would like to see whether DiffS4L is still helpful in the presence of a language model. To this end, we introduce a 4-gram language model to the English ASR task. As can be observed from the rows marked with ‘4-gram’ in Table 4.7, not only does the performance advantage persist when the 4-gram language model is introduced, but also the gap is largely the same as that without the language model. These results verify the robustness of DiffS4L regardless of the use of the language models.

Size of Finetuning Dataset To study the impact of the size of the finetuning dataset on performance, we finetune SSL models on the train-clean-100 subset of LibriSpeech-960 dataset and compare the results to those obtained from the 10-hour supervision set of LirbriLight. We observe that in the 100-hour low-resource setting (EN-100-100) Wav2vec2-DiffS4L systems still have a relatively large gain compared to the baseline

Wav2vec-Real. In the high-resource setting (EN-960-100) where there is a sufficient amount of labeled speech, the gain diminishes.

4.4.12 Full ASR results on MLS and CommonVoice

We provide the full ASR results on MLS and Commonvoice dataset in Table 4.7. To better examine the robustness of DiffS4L under different settings, we perform some additional experiments on the MLS ASR task (Wav2vec-SS/NS in Table 4.7).

Additional Test Set The MLS and Commonvoice datasets come with a dev set and a test set for each language, both of which can be utilized as test sets to evaluate the ASR performance. In the main paper, we reported the dev set performance. Here, we include the results on the test set to show the statistical significance of the performance advantage of DiffS4L. As shown in the columns under 'Test' in Table 4.7, DiffS4L maintains a consistent advantage over the baseline, which is trained on the 100 hours of real speech alone, and the performance gaps are similar to that in the dev set. These results confirm the significance of the benefit induced by DiffS4L-generated data.

Dataset Compositions In the main results, we only examined the effect of varying dataset compositions for English ASR. In this section, we extend the experiment to different languages by introducing the Wav2vec-SS/NS, which are trained on the 100+860+0 dataset composition, *i.e.*, without NC speech. As can be observed from the rows marked with 'Wav2vec-SS/NS' in Table 4.7, the performance always deteriorates when NC speech is removed. This is a rather impressive finding because different languages have different structures, some of which are easier to capture than others. The fact that NC speech is able to improve the performance for *all* these languages indicates that the diffusion model can successfully capture all the different types of structural information.

Table 4.6: Full ASR results on LibriSpeech English dataset, including the CER/WER of Wav2vec2 model pretrained on 100/960 hours and fine-tuned on 10/100 hours. Results of Wav2vec-Aug are included for EN-100-10 experiment for a comparison with Wav2vec2 and Wav2vec-DiffS4L.

		Dev-clean		Dev-other		Test-clean		Test-other	
Model	LM	CER	WER	CER	WER	CER	WER	CER	WER
EN-100-10									
Wav2vec-Real	None	7.13	22.17	15.06	37.57	7.17	22.62	15.74	39.24
	4-gram	9.79	19.91	18.45	36.05	9.80	20.20	19.40	37.71
Wav2vec-Aug	None	6.92	22.06	14.83	37.17	6.95	22.48	15.64	39.01
	4-gram	9.24	19.36	18.28	36.02	9.47	19.64	19.22	37.35
Wav2vec-SS/NS	None	5.58	17.43	12.84	32.58	5.59	17.78	13.31	33.99
	4-gram	7.84	15.41	15.92	31.26	7.91	15.74	16.47	32.54
Wav2vec-DiffS4L	None	5.19	16.67	11.85	30.03	5.31	17.39	12.17	31.27
	4-gram	7.70	15.00	14.99	28.73	7.57	15.01	15.40	29.91
EN-960-10									
Wav2vec-Real	None	3.18	10.49	6.69	18.03	3.07	10.39	6.64	18.53
	4-gram	5.17	9.14	9.35	16.98	5.1	9.06	9.31	17.43
Wav2vec-DiffS4L	None	2.98	9.93	6.31	17.19	3.03	10.14	6.27	17.55
	4-gram	4.97	8.42	8.80	15.91	5.08	8.76	8.92	16.36
EN-100-100									
Wav2vec-Real	None	4.44	13.95	14.43	34.47	4.56	14.60	15.50	36.90
	4-gram	6.42	12.25	17.62	33.44	6.67	12.87	18.81	35.75
Wav2vec-DiffS4L	None	2.93	9.56	10.51	25.94	3.03	9.98	10.74	26.77
	4-gram	4.81	8.22	13.32	24.76	5.02	8.69	13.75	25.91
EN-960-100									
Wav2vec-Real	None	1.65	5.60	5.03	13.62	1.65	5.74	4.76	13.4
	4-gram	3.33	4.60	7.33	12.77	3.43	5.07	6.98	12.49
Wav2vec-DiffS4L	None	1.61	5.58	4.80	12.91	1.63	5.66	4.63	12.93
	4-gram	3.33	4.60	7.19	12.26	3.41	5.01	7.02	12.16

Table 4.7: ASR performance of Wav2vec2 pretrained on DiffS4L-generate data on LibriSpeech, MLS and CommonVoice dataset

	Dev		Test		Dev		Test	
Model	CER	WER	CER	WER	CER	WER	CER	WER
	EN				**PO**			
Wav2vec-100R	7.13	22.17	7.17	22.62	13.83	45.75	16.48	50.92
Wav2vec-SS/NS	5.58	17.43	5.59	17.78	10.37	35.17	12.45	40.16
Wav2vec-DiffS4L	5.19	16.67	5.31	17.39	9.88	34.60	11.96	39.78
	DE				**BA**			
Wav2vec-100R	8.33	30.44	9.93	33.83	10.16	43.81	11.82	47.99
Wav2vec-SS/NS	6.67	24.48	7.96	27.45	-	-	-	-
Wav2vec-DiffS4L	6.37	23.27	7.55	26.11	8.90	37.07	9.12	37.09
	ES				**CKB**			
Wav2vec-100R	7.10	27.22	7.08	27.33	7.23	39.04	7.75	40.86
Wav2vec-SS/NS	6.20	23.46	6.29	23.44	-	-	-	-
Wav2vec-DiffS4L	4.49	16.65	4.48	16.83	6.71	29.70	6.48	26.65
	FR				**CY**			
Wav2vec-100R	16.16	45.50	14.49	41.84	20.58	62.05	17.25	49.37
Wav2vec-SS/NS	12.12	35.80	10.61	31.65	-	-	-	-
Wav2vec-DiffS4L	11.91	34.77	10.61	31.13	16.70	52.28	12.48	37.45
	IT				**MHR**			
Wav2vec-100R	8.33	35.08	7.80	33.62	10.74	45.41	12.91	49.43
Wav2vec-SS/NS	8.09	34.39	7.35	32.10	-	-	-	-
Wav2vec-DiffS4L	6.24	27.22	5.54	24.43	9.44	37.52	10.04	39.19
	NL				**SW**			
Wav2vec-100R	17.83	50.92	11.55	39.09	8.80	31.54	8.83	29.71
Wav2vec-SS/NS	15.31	46.78	9.49	33.85	-	-	-	-
Wav2vec-DiffS4L	14.69	44.83	9.37	33.25	6.99	25.92	7.55	24.55
	PL				**TA**			
Wav2vec-100R	11.42	44.22	9.92	43.20	9.16	47.20	11.19	54.07
Wav2vec-SS/NS	7.80	32.75	7.67	35.72	-	-	-	-
Wav2vec-DiffS4L	7.14	30.95	7.56	34.90	7.51	40.98	8.58	45.17

Table 4.8: ASR performance across the number of clusters for speech units.

#Clusters	100	200	300	500
CER/WER	32.9/51.0	27.9/43.7	28.3/42.2	**23.0/34.1**

4.4.13 Number of Clusters for Speech Units

In the preliminary experiment, we train a WaveNet conditioned on the speech units of 100 cluster, 200 clusters, 300 clusters and 500 clusters, to synthesize the English speech. We then measure the quality of synthesized speech using a Wav2vec2-CTC model. The results are show in Table 4.8. The 500-cluster speech units yield the best ASR performance, indicating the 500-cluster units better capture the speech information.

In addition, we perform the ABX test from Zero Resource Speech Challenges 2020 [154, 155] on the 500-cluster units and get the ABX within/across speaker score of 7.87/10.29, which is not too far away from the 200-cluster 'hubert_l6' units reported in [155], which has an ABX score of 5.99/7.31.

4.4.14 Masking Length

Recall that the NC data is generated by conditioning on $\boldsymbol{R}_0$ with 80% frames masked out, as shown in Figure 4.3(d). We would like to investigate whether the masking length has an impact on the performance. We thus retrain two partially-conditional diffusion models, one with 50% masking length and the other with 100% (which becomes totally unconditional). We then generate two synthetic datasets, whose compositions are both 100+430+430, but whose NC data are generated with 50% and 100% masking length, respectively. The corresponding Wave2vec English ASR results are shown in Figure 4.6. As shown, there are only slight differences in the performance, with the optimal achieved by 80% masking length. We conjecture that two factors influence the performance when changing the mask length. One is the amount of novel content, which increases as masking length increases; the other is the quality of the generated speech, which tends to decrease as masking length increases. Therefore, pushing the mask length to both extremes negatively impact the performance.

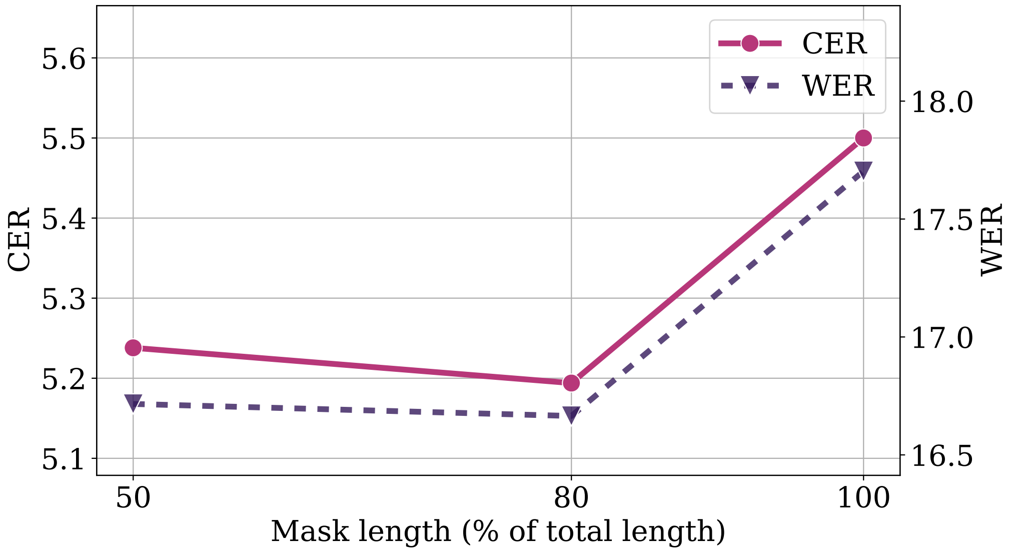


Figure 4.6: Performance over different masking ratios when synthesizing NC speech.

4.4.15 Additional Experiment using EDM

We experiment with another diffusion model EDM [156], instead of DDPM to generate synthetic data. The dataset configurations and evaluation procedures are exactly the same as described in Section 4.4, except that the diffusion process is changed. The architecture of the speech synthesizer remains the same while the diffusion training and inference pipeline follow the official implementation of EDM[5]. We keep the default hyperparameters of the original EDM implementation except for the data standard deviation, which is calculated from our training data. The diffusion model is trained for 300k iterations on eight V100-SXM2-32GB GPU with a batch size of 32 per GPU and a learning rate of 5×10^{-4}. We use adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ and inverse square root scheduler with 32000 warmup updates. The sampling process for data generation uses 18 steps.

The results of Wav2vec-DiffS4LEDM and HuBERT-DiffS4LEDM trained using synthetic data are shown in Table 4.9. We get similar ASR and SUPERB performances as using DDPM, suggesting that the diffusion models consistently generate babble that better captures the inherent speech

[5] `https://github.com/NVlabs/edm`

Table 4.9: Results of EDM on (a) English automatic speech recognition and (b) SUPERB benchmark. The Wav2vec-960R and HuBERT-960R are topline models.

	(a) English ASR		(b) SUPERB								
			KS	IC	SID	ER	Qbe	SF		ASV	SD
Task/Metric	CER↓	WER↓	ACC↑	ACC↑	ACC↑	ACC↑	MTWV↑	F1↑	CER↓	EER↓	DER↓
Wav2vec-DiffS4LEDM	5.20	16.81	92.99	93.94	47.28	61.24	0.0327	81.66	35.15	7.88	7.30
HuBERT-DiffS4LEDM	5.21	17.03	94.55	95.94	43.78	61.80	0.0501	82.68	34.32	7.42	7.26

structure.

4.5 Summary

In this study, we examined SSL from an information efficiency perspective and found that performance can be greatly improved by utilizing the information present in the pretraining dataset, particularly in low-resource settings. We discovered that synthetic data is an effective way to extract information and enhance SSL performance. Specifically, diffusion models were found to be particularly capable of capturing complex structures in speech that traditional pretraining methods cannot; thus even synthetic babbles contain valuable information for SSL training. DiffS4L opens the door to a new approach to speech SSL. One limitation of DiffS4L is that it is a time-consuming process, as it involves training of multiple networks sequentially. As a next step, we plan to investigate more efficient methods of information sharing between diffusion models and SSL models to reduce the need for synthetic data generation and prolonged pretraining.

CHAPTER 5

UNSUPERVISED TEXT-TO-SPEECH SYNTHESIS BY UNSUPERVISED AUTOMATIC SPEECH RECOGNITION

5.1 Introduction

Text-to-speech (TTS) synthesis is an essential component of a spoken dialogue system. While being capable of generating high-fidelity, human-like speech for languages such as English and Mandarin, the existing state-of-the-art TTS systems such as Tacotron 1&2 [157, 15], Deep Voice 3 [158], FastSpeech [159] and TransformerTTS [160] are trained with a large amount of parallel speech and textual data. The reliance on a large amount of transcribed speech makes such systems impractical for the majority of the languages in the world. Training a supervised text-to-speech (TTS) system requires dozens of hours of single-speaker high-quality recordings [161], but collecting a large amount of single-speaker, clean, and transcribed speech corpus can be quite time-consuming and expensive [162]. A potential way to relax such a requirement is to use *non-parallel* untranscribed speech and text corpora in the same language. Such corpora are much easier to obtain in practice since no human annotators are required in the data collection process, thanks to the abundance of text data on the Internet. Learning to perform TTS using non-parallel speech and text, or *unsupervised TTS*, poses unique challenges: first, standard supervised training criteria, such as autoregressive mean-squared error, are no longer applicable; further, to learn the latent alignment between the spoken frames of an utterance and its transcript, the model now needs to search over every utterance and every transcript in the entire corpus instead of limiting the search space within a single utterance-transcript pair. As a result, instead of learning to memorize the correspondence between text and speech, an unsupervised TTS model needs to decompose and generalize information shared by the two sources to reduce the search space.

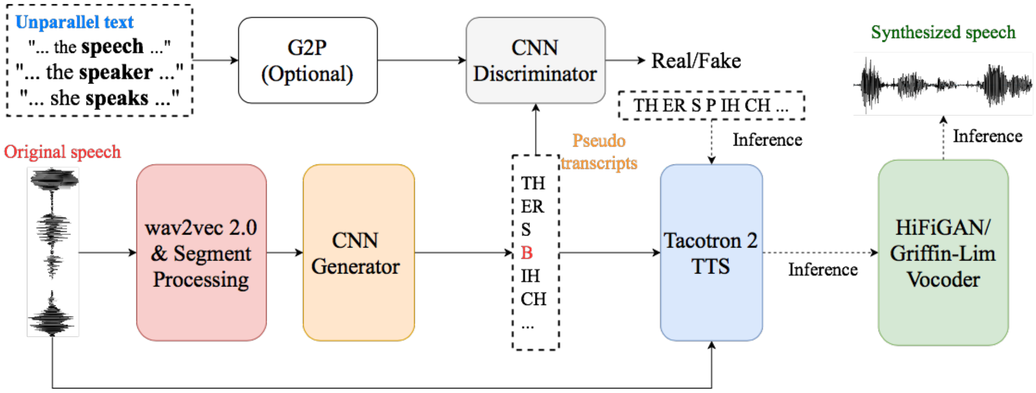


Figure 5.1: Network architecture for unsupervised speech synthesis

This paper proposes the first model for solving the unsupervised TTS problem. We[1] decompose training the model into two tasks, one unsupervised ASR task and one supervised TTS task. The second supervised TTS task uses pseudo-transcripts obtained by solving the first task. Our model takes advantage of Wav2vec-U [163], the best publicly available unsupervised ASR system, and can generalize seamlessly to future updates of the Wav2vec-U model. We conduct our unsupervised TTS experiments on seven languages. We further provide an in-depth analysis of the effect of several components on unsupervised TTS performance, including the grapheme-to-phoneme (G2P) converter and the vocoder.

5.2 Related works

Several recent works have attempted to develop TTS systems for low-resource scenarios. One direction of research is to replace ground truth phoneme or grapheme labels required for supervised TTS with other units obtained with less or no supervision, such as articulatory features [164], or acoustic units discovered by self-supervised speech representation models such as vector-quantized variational auto-encoder (VQ-VAE) [165, 166] and HuBERT [167, 168, 169]. The unsupervised, textless approach can be applied to any language, including those without any written form. However, the per-

[1] The project described in this chapter is part of [3], with co-authors Junrui Ni, Liming Wang, Yang Zhang, Kaizhi Qian, Shiyu Chang, and Mark Hasegawa-Johnson. The samples generated by our models are available in `https://cactuswiththoughts.github.io/UnsupTTS-Demo`. The code is available in `https://github.com/lwang114/UnsupTTS`.

formance of such a system is limited by the quality of the acoustic units used, which can be quite noisy due to the difficulty of acoustic unit discovery. To address this limitation, [170, 171, 172, 173] have studied the use of other sensory modalities such as images in place of textual transcripts as a weaker form of supervision for conditional generation of speech, or “TTS without T”, using various attention mechanisms over the visual features. Another approach to address this issue is to allow a small amount of transcribed speech and train the TTS in a semi-supervised fashion [174, 161]. Specifically, [174] leveraged unpaired speech and text data by constructing pseudo-corpora via dual transformation between ASR and TTS systems with on-the-fly refinement followed by knowledge distillation, while LRSpeech [161] trained an ASR and a TTS system that used only several minutes of paired single-speaker, high-quality speech for TTS, and several hours of low-quality, multi-speaker data for ASR.

Our approach relies on the most recently published unsupervised automatic speech recognition (ASR) system [163], which learns to recognize phones by leveraging one-hot phone sequences from an unpaired text corpus. Earlier works on unsupervised ASR typically try to match the empirical prior and posterior distributions of phonemes either using cross-entropy [175] or adversarial loss [176]. Using powerful large-scale, self-supervised, pre-trained acoustic features such as Wav2vec2 [177] and a generative adversarial network (GAN) based system, the adversarial approach achieves comparable performance to its supervised counterpart on large-scale speech datasets for multiple languages [163].

5.3 Proposed method

The proposed unsupervised TTS system contains two stages: training an unsupervised ASR system and training a supervised TTS system. We first evaluate the proposed unsupervised TTS system in English. To examine the generality of this method in other languages, we also train the two-stage system in Hungarian, Spanish, Finnish, German, and Japanese.

5.3.1 Unsupervised ASR with Wav2vec-U

An unsupervised ASR system is trained with unpaired speech and text. We use Wav2vec-U [163] as the unsupervised ASR system. The Wav2vec-U system follows a two-step approach: GAN training and self-training. In the GAN training step, a simple 1-layer CNN acts as the generator, which takes the segment representations extracted from a pre-trained Wav2vec2 model [177] and outputs a sequence of distributions over text units, where consecutive segments with the same argmax value are collapsed. The discriminator, a 3-layer CNN, tries to tell which source (real or generated) the input sequence is from, and the generator is trained against the discriminator. This is achieved by iteratively maximizing the likelihood of the generated phoneme sequence to train the generator and minimizing the binary cross-entropy loss to train the discriminator.

In addition, since GAN training can be very unstable, we search over the weights for regularization losses such as gradient penalty loss, segment smoothness penalty, and phoneme diversity loss as described in [163]. We also validate the model with 50-100 transcribed utterances from the corpus to ensure convergence instead of using the unsupervised metric as described in [163]. After GAN training, greedy decoding is applied to the generator's output over the training set. We then train a hidden Markov model (HMM) with framewise speech representations extracted from a Wav2vec2 model as input and pseudo-text decoded by the generator as output. Finally, we decode the entire corpus again using the newly-trained HMM to obtain pseudo-transcripts for the supervised TTS system. Except for English, we opt not to further finetune a Wav2vec2 model with the pseudo-transcripts from the HMM.

5.3.2 Supervised TTS with Tacotron2

A supervised TTS system takes the pseudo-transcripts from the unsupervised ASR system and outputs mel-spectrograms. A modified Tacotron2 [15] is used with an additional guided attention loss [178]. We perform an unsupervised model selection process by feeding the model with pseudo-transcripts instead of ground truth transcripts when computing validation loss. During the evaluation, ground truth transcripts are used as inputs to the TTS.

Character error rates (CER) and word error rates (WER) are used to measure how much linguistic content is preserved by the TTS. We train a fully supervised TTS system using real text instead of pseudo-text and calculate the CER and WER on the same subset for a meaningful comparison. To obtain the CER and the WER on each language, we either directly use a publicly available Wav2vec2 speech recognizer (for English) or finetune a pre-trained Wav2vec2 model on each language individually.

5.4 Experiments

5.4.1 Unsupervised TTS on English

We first evaluated the two-stage unsupervised TTS system on English. To train the first-stage Wav2vec-U system, we used speech utterances from the 24-hour single-speaker LJSpeech corpus [179] and text samples from the LibriSpeech language modeling corpus [73]. We set aside about 300 utterances for validation and about 500 utterances from the LJSpeech corpus for testing. We kept the ground truth transcripts for validation and test sets and used the rest for training without ground truth transcripts. The speech representations were extracted using a publicly available Wav2vec2 Large model trained on LibriLight [151], and the segment representations were built following the pre-processing procedures in [163].

Table 5.1: Unsupervised ASR results on the LJSpeech dataset using English Wav2vec2 pre-trained features

Language	Duration (hr)	Unsup ASR (PER)	
		No ST	ST
English	24	12.37	3.59

The non-parallel text samples used for training, as well as the ground truth transcripts for the validation and test utterances, were converted to phones using a grapheme-to-phoneme (G2P) converter [180]. The best weights for the auxiliary penalties of the Wav2vec-U system, *i.e.*, code penalty, gradient penalty, and smoothness weight, *c.f.* [163], were determined by grid search, and we chose the best model based on its PER over the validation

Table 5.2: Unsupervised TTS results on the LJSpeech dataset using English Wav2vec2 pre-trained features

Language	Unsup TTS		Supervised TTS	
	CER	WER	CER	WER
English	4.56	11.95	3.93	10.76

set after 150k steps with a batch size of 160. Wav2vec-U GAN training is sometimes unstable in ways that we could only detect by using 50-100 supervised validation examples, which were the only places during training where we used paired data. The results of this stage is shown in Table 5.1. After determining the best Wav2vec-U model, its output phone sequence was then refined using a self-training (ST) process [163] as follows. First, we used framewise Wav2vec2 features after PCA transformation as input and pseudo phone sequences transcribed by the Wav2vec-U generator as targets to train a triphone HMM. The triphone output from the HMM was decoded into words with an HCLG decoding graph, and we further fine-tuned a Wav2vec2 Large model using the pseudo character targets obtained from the above step, under the Connectionist Temporal Classification (CTC) loss [4]. Both steps were validated with the corresponding pseudo-text for the validation set. As shown in Table 5.1, ST reduces the phone error rate on the test set by 70% relative and provides very accurate transcripts for the second-stage TTS system. We used the Fairseq toolkit [150] to train the GAN of Wav2vec-U and used the Kaldi toolkit [181] to train the triphone HMM and to build the decoding graph.

To train the second stage, we used the Tacotron2 [15] model implemented by ESPnet [182] as the TTS component of the system. The ESPnet Tacotron2 follows the original Tacotron2 model, except that another guided attention loss [178] was calculated on top of the encoder-decoder attention matrix to ensure that the attention matrix was not too far from diagonal. During training, the Tacotron2 system takes pseudo phone transcripts as inputs. These pseudo phone transcripts are converted by G2P from the word-level hypotheses generated by the fine-tuned Wav2vec2 model (in the final step of Wav2vec-U training). The outputs of the Tacotron2 system are 80-dimensional mel-spectrograms. The TTS component was trained for 80 epochs, with the same validation and test splits as the Wav2vec-U

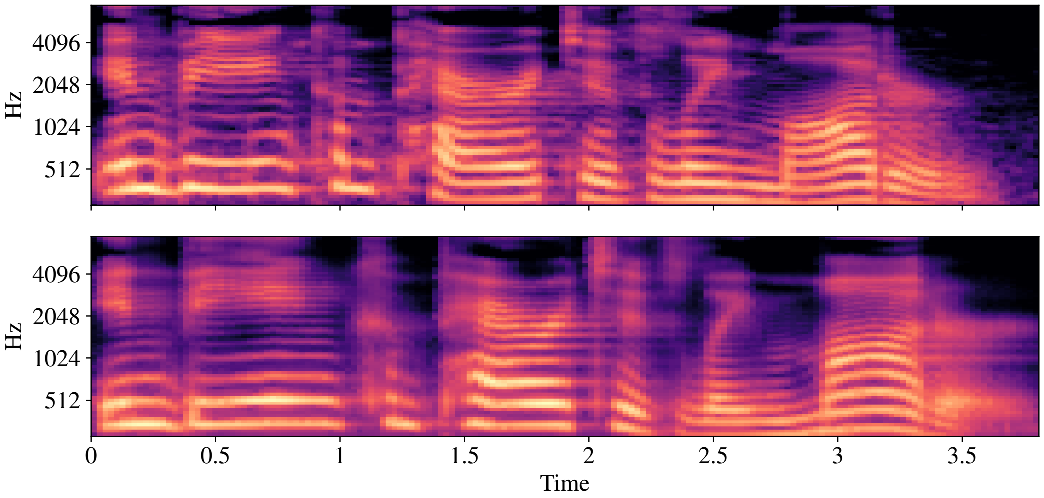


Figure 5.2: Mel-spectrograms for ground truth (upper) and synthetic speech by the unsupervised TTS model (lower) for the English sentence "in being comparatively modern."

system. During validation of the Tacotron2 model, we calculated the reconstruction loss based on pseudo-text instead of real text. During testing, we fed the trained TTS component with real, phonemicized text transcripts for the test set to obtain mel-spectrograms from Tacotron2 and synthesized raw audios with HiFiGAN [183]. We calculated the CERs and raw WERs without additional language models using a publicly available Wav2vec2 Large model fine-tuned on LibriSpeech. Table 5.2 shows the two error rates on the synthesized test utterances using our proposed unsupervised system (Unsup TTS). Compared with another fully-supervised Tacotron2 model trained and validated with real, phonemicized text transcripts, our unsupervised system only lags behind 0.63% absolute in terms of CER and 1.19% absolute in terms of WER. Figure 5.2 plots the mel-spectrogram of a synthetic speech example by our unsupervised model, which shows that except for the temporal patterns, the mel-spectrogram by the unsupervised TTS looks very similar to the ground truth with very little loss of linguistic content.

Table 5.3: Unsupervised ASR results on the CSS10 dataset using English Wav2vec2 pretrained features

Language	Duration (hr)	Unsup ASR (CER)	
		No ST	ST
Japanese	15	26.12	17.81
Hungarian	10	25.08	15.26
Spanish	24	20.80	14.57
Finnish	10	29.78	21.00
German	17	26.31	19.47
Dutch	14	45.65	39.24

Table 5.4: Unsupervised TTS results on the CSS10 dataset using English Wav2vec2 pretrained features

Language	Unsup TTS		Supervised TTS	
	CER	WER	CER	WER
Japanese	17.98	47.81	17.87	36.23
Hungarian	27.78	76.82	18.05	63.14
Spanish	23.03	55.52	18.19	36.74
Finnish	36.05	84.46	22.84	58.67
German	17.25	56.78	11.28	40.94
Dutch	53.01	89.41	34.53	76.71

5.4.2 Unsupervised TTS on CSS10 Languages

We evaluated our unsupervised TTS system on six additional languages: Japanese, Hungarian, Spanish, Finnish, German and Dutch from the CSS10 dataset [184]. The total duration of each language is listed in Table 5.3. The experiments followed the same steps as the English experiment in Sec 5.4.1. We used the same English Wav2vec2 Large model to extract speech representations, the same training pipeline to train the Wav2vec-U system, followed by self-training, on the extracted audio features, and the Tacotron2 model for TTS. The unsupervised ASR results are shown in Table 5.3. There were still a few differences in details in this multilingual experiment. While the previous English experiment used audio and text from different datasets, due to resource limits, these multilingual experiments used a potentially easier setting where both the audio and text were drawn from the same CSS10 dataset with their paired relationship broken up. We did not convert the

Table 5.5: The effect of different pretrained vocoders (Griffin-Lim, HiFiGAN) on unsupervised TTS results for LJSpeech and various languages from CSS10

Language	Griffin-Lim		HiFiGAN	
	CER	WER	CER	WER
English	5.02	12.83	**4.56**	**11.95**
Japanese	**17.98**	**47.81**	20.58	54.09
Hungarian	27.78	76.82	**26.92**	**76.60**
Spanish	**23.03**	**55.52**	29.41	68.82
Finnish	**36.05**	**84.46**	37.66	87.48
German	**17.25**	**56.78**	18.45	59.90

Table 5.6: The effect of different text units on unsupervised TTS using Griffin-Lim vocoder

Language	Phoneme		Grapheme	
	CER	WER	CER	WER
Hungarian	**22.73**	**68.80**	27.78	76.82
Finnish	**27.58**	**67.87**	36.05	84.46
Dutch	**22.04**	**56.85**	53.01	89.41

graphemes into phonemes but instead directly used the characters in each language to train and evaluate the model. We split the audio and text data into train and validation sets with a ratio of 99 to 1, which gave us about 50 to 100 validation utterances depending on the dataset size. The self-training step of the first stage only contained a character-based HMM (instead of a triphone HMM with HCLG decoding) for generating pseudo labels, and we did not have a second step of fine-tuning a Wav2vec2 model as in the English experiment. During the evaluation in the second stage, we used a Griffin-Lim vocoder to synthesize the audios from the mel-spectrogram generated by Tacotron2, and the results reported in Table 5.4 were calculated using audios from the Griffin-Lim vocoder instead of the HiFiGAN vocoder. We switched to the Griffin-Lim vocoder because we empirically found that it yielded lower error rates on these languages. To calculate CER and raw WER, we fine-tuned a publicly available Wav2vec2 Base model for each language individually, using paired speech and character-level transcripts from each CSS10 corpus.

The multilingual results in Table 5.4 confirm the conclusions we reach in the English experiments. Although the self-training step is simplified to only a character-based HMM in CSS10 multilingual experiments, the self-training step still greatly reduces the error rates by 25% to 40% relative to all the languages. Compared to the fully-supervised TACOTRON2 model trained using real text transcripts, the CERs of our unsupervised systems differ from those of the supervised counterparts by only about 9% absolute on average while requiring only a few paired utterances during validation. Further, we observe that the gap in WER between supervised and unsupervised TTS systems generally is about 10-20% absolute for all languages except Finnish, a much larger gap than CER. We hypothesize that it may be due to the lack of a robust language model in the TTS systems, making it harder for the model to preserve word-level information when training with the noisy (pseudo-) transcripts. Last but not least, we observe that the unsupervised ASR performance does not always limit the performance of unsupervised TTS. In the case of German, the TTS trained with pseudo-transcripts achieves a *lower* CER compared to the unsupervised ASR system, which suggests that the TTS has some internal mechanism to correct the noise in the pseudo-transcripts provided by the ASR.

5.4.3 Comparison Between GRIFFIN-LIM and HIFIGAN

A comparison between the error rates of using GRIFFIN-LIM and HIFIGAN vocoders is presented in Table 5.5. We observe that the GRIFFIN-LIM vocoder yields lower CERs and WERs than the HIFIGAN vocoder in all languages except English and Hungarian, even though informal listening suggests that HIFIGAN generates more natural speech with fewer artifacts. We hypothesize that HIFIGAN works better for English because it is pretrained on the English LJSPEECH dataset and may not generalize very well when applied to datasets of different languages.

5.4.4 Comparison Between Phoneme and Grapheme

We trained additional phoneme-based unsupervised TTS models in Hungarian, Finnish, and Dutch to study how the text units affect the TTS performance. The training procedure was the same as that described in Sec 5.4.2, except that in the very beginning, we converted the language-specific graphemes to the phonetic annotations, i.e., the International Phonetic Alphabet IPA, using LanguageNet G2Ps [102]. The CERs and WERs are reported in Table 5.6. The table shows that the phone-based systems yield significantly lower error rates than the grapheme systems. As graphemes are the smallest functional unit of a writing system, it involves extra complexity on top of the phone systems. Thus, modeling the grapheme systems is harder than modeling the phone systems, as indicated by its higher error rates. The gap between grapheme and phoneme systems is considerably smaller for Hungarian and Finnish than for Dutch. One probable explanation is that spelling and phonetic transcription is far more regular for the former two languages than for Dutch.

5.5 Summary

In this work, we designed a two-stage system for training an unsupervised TTS system without paired data. Our systems do not require paired speech and text during training except for a small validation set to ensure the convergence. The final unsupervised TTS system demonstrates competitive intelligibility in English and a slight degradation in intelligibility in six other languages on the level of supervised TTS models. We further show that phonemes work better than graphemes as text units for our systems. In the future, we would like to explore unsupervised TTS with truly non-parallel datasets for languages other than English and ways to improve the stability of the unsupervised ASR system in the first stage of our system.

CHAPTER 6

LANGUAGE EMBEDDING FOR ZERO-SHOT CROSS-LINGUAL TRANSFER

6.1 Introduction

Modern end-to-end neural network based speech recognition systems (ASR) have achieved great success on resource-rich languages such as English and Mandarin [8]. However, most existing languages are resource-deficient, making it hard for neural networks to achieve similar accuracy.

Multilingual and Cross-lingual phonetic recognition attempt to partially solve the low-resource problem by building a universal phone recognizer that transcribes speech from different languages into corresponding phone sequences, under the assumption that there exists a universal acoustic model shared by all languages. If this assumption holds, an ideal recognizer should have low error rates on not only the languages it is trained on, *i.e. multilingual error rates*, but also the unseen languages, *i.e. cross-lingual error rates*, in a zero-shot setting.

However, although multilingual training is shown to improve the performance on seen languages [185, 78, 76], it does not greatly benefit zero-shot generalization to unseen languages [75]. This implies that acoustic models implicitly captured in these multilingual systems are language-specific, and thus would not generalize to unseen languages unless additional information about the unseen languages is supplied.

Motivated by this, we[1] propose to improve the zero-shot cross-lingual recognition accuracy by incorporating a language embedding that captures two types of external knowledge – *phylogenetic similarity* and *phone inventory*. For phylogenetic similarity, we extract phylogenetic information from Glottolog [186], which is a large graph specifying the belonging relations

[1] The project described in this chapter is part of [77], with co-authors Junrui Ni, Yang Zhang, Kaizhi Qian, Shiyu Chang, and Mark Hasegawa-Johnson. The code is available in `https://github.com/Hertin/zeroshot_langemb`

between nodes of dialects, languages, and language families. Assuming the closeness of the two languages in the graph captures the phylogenetic similarities between the languages, we use Node2vec [187] to extract vector representations for each node. For the phone inventory information, we extract a binary vector to represent the phoneme inventory for each language from Phoible [188]. The two vectors are combined and fed into a language encoder and produce the language embedding, on which the multilingual phoneme classifier is conditioned. The phone inventory information is also imposed by masking on the output logits with the binary vector.

The experiments show that the proposed algorithm with language embedding and masking improves the performance over the baselines on the unseen languages in the zero-shot setting by a large margin (4%–8% absolute). Ablation study shows that both the phylogenetic and phone inventory information are crucial for performance improvement.

6.2 Related Works

There has been active research on multilingual recognition. A large number of languages do not have enough parallel speech and text data, and deep learning models trained on these languages usually have high error rates [75]. Multilingual speech recognition mitigates the data sparsity by training the network on a combined dataset from several languages. The network usually has a common encoder that extracts acoustic information from audio features and can either have a common decoder with a shared phoneme inventory [76] or language-specific decoders with private phone [78, 79, 80] or character inventories [81, 82, 83]. Multilingual ASR can benefit from the use of self-supervised pretraining algorithms such as contrastive predictive coding [37, 10, 11], which pretrains a model on large amounts of unlabeled raw audio data to predict neighboring frame representations given the center frame. Multilingual models generally have better accuracy and robustness compared to monolingual models [75, 76, 78, 79, 80], as they benefit from increased amount and diversity of data.

Language or dialect embedding that models the language-dependent or dialect-dependent biases has been shown to improve multilingual ASR systems [84, 85, 86, 87]. The embedding can be a one-hot vector specifying language ID [84, 86] or a vector learned from acoustic data under a standard multilingual model [85, 87] and can be used as additional input features to the network [84, 86], as adapter modules for language-specific adjustments [86] or as interpolation weights for the encoder [85]. However, the embeddings in all these previous works depend on the test language being either one of the training languages (in the case of a one-hot embedding) or recorded in a fashion that makes its acoustic embedding vector a useful predictor of its phoneme-to-sound acoustic models.

However, studies have found that multilingual models do not generalize well to unseen languages [75], without adapting to parallel data from that language. While multilingual training can yield error rates 10–20% below monolingual training, the leave-one-out cross-lingual error rate when applying the multilingual model to an unseen language can be 70–90%. Because of the high error rates of zero-shot cross-lingual ASR, most researchers studying cross-lingual ASR have chosen pragmatically to define that term to mean few-shot rather than zero-shot recognition, *e.g.*, by finetuning using one hour [189, 190] or a few hours [191] of transcribed data in the target language. Perhaps the prior work most similar to the work in this paper is a set of experiments using the PHOIBLE [188] phoneme inventory of a language to define an untrained, knowledge-based linear output layer called the "signature matrix" [192, 76]; our phone token masking strategy is a simplification of the signature matrix, and our proposed language encoding is an enrichment of the same.

6.3 Methods

Previous works have shown that it is hard to achieve good performance on zero-shot cross-lingual recognition without any knowledge of the testing language. We, therefore, consider incorporating extra information about the testing language. Figure 6.1 shows the overview of the proposed architecture. The proposed system is a CTC+Attention system based on [75], with three additions: (1) WAV2VEC2-based feature extraction based on [10], (2) phoneme inventory masking similar to [76], and (3) the proposed typology-based language encoder.

The language encoder The language encoder includes two sets of in-

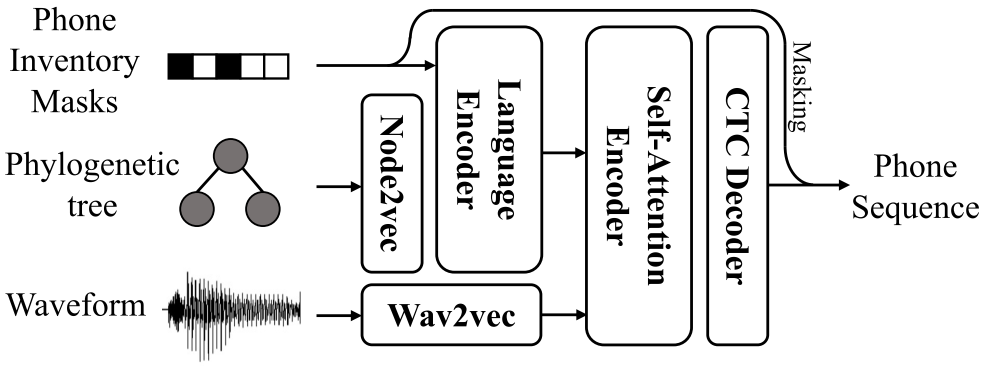


Figure 6.1: Architecture overview

formation about the test language. The first is the language phylogenetic information extracted from GLOTTOLOG, which is a graph containing dialects, languages, and language families as nodes and the belonging relationships as edges. We use NODE2VEC [187] to embed the nodes so that the languages that are close in the graph have larger cosine similarities.

Similar to the multilingual allophone system in [76], we also include phone inventory information from PHOIBLE [188], a cross-linguistic phonological inventory database for over 2000 distinct languages. We combine inventories for all the languages to create a shared phoneme inventory and use a binary vector to represent the phoneme set of each language.

The language node embedding and the binary phoneme inventory vector are concatenated, forming a general representation applicable to at least 2,000 languages. The vector is then fed into the language encoder, producing a language embedding as an additional input to the phoneme classifier.

Wav2vec2 Feature Extraction Considering the remarkable performance boost brought by pretrained unsupervised acoustic representation, we experiment on the feature extractor (referred to as feature encoder in [11]) from WAV2VEC2[2] that is pretrained on 1000 hours of LIBRISPEECH [73].

Phone Inventory Masking In addition to feeding the phone inventory asks as an input to the language encoder, we also directly use it to mask out the non-existing phonetic tokens in the output layer, which has been shown to be effective in reducing the error rate, especially for unseen languages.

[2] `https://github.com/pytorch/fairseq/tree/master/examples/wav2vec`

Table 6.1: Sources of data used in our cross-lingual experiment. The upper part is the training languages and the lower part is the testing languages. Type column denotes whether the corpus contains spontaneous (Sp.) or read speech. Len column shows the total duration of all utterances in hours. Family column shows the language family.

Language	Abbr	Corpus	Type	Family	Len
Bengali	103	Babel	Sp.	Indo-Aryan	215
Vietnamese	107	Babel	Sp.	Vietic	215
Zulu	206	Babel	Sp.	Bantu	211
Amharic	307	Babel	Sp.	Ethiopic	204
Javanese	402	Babel	Sp.	Austronesian	204
Georgian	404	Babel	Sp.	Kartvelian	190
Dutch	N	CGN	Read	Germanic	64
Czech	CZ	GP	Read	West Slavic	29
French	FR	GP	Read	Romance	25
Mandarin	CH	GP	Read	Sinitic	31
Thai	TH	GP	Read	Tai	22
German	GE	GP	Read	Germanic	18
Portuguese	PO	GP	Read	Romance	26
Turkish	TU	GP	Read	Turkic	17
Bulgarian	BG	GP	Read	South Slavic	21
Cantonese	101	Babel	Sp.	Sinitic	215
Lao	203	Babel	Sp.	Tai	207
Croatian	CR	GP	Read	South Slavic	16
Spanish	SP	GP	Read	Romance	22
Polish	PL	GP	Read	West Slavic	24

6.4 Experiment Setup

6.4.1 Dataset

The performance of our model is evaluated on a corpus that consists of 20 languages, 8 from IARPA Babel project corpora, 1 from CGN (Spoken Dutch Corpus) [193] and 11 from GlobalPhone [98] (GP), as summarized in Table 6.1. We only use the read speech part of CGN corpus. We use the default 8:1:1 train-dev-test partition provided by Babel corpora and split CGN and GlobalPhone corpora into 8:1:1 partitions with non-overlapping speakers. Since our task is cross-lingual phonetic token recognition, the train and dev partitions of the testing languages are not used. We select 5 languages,

Table 6.2: Phonetic token error rates (PTER) in percentage. The rows from "103" to "BG" are PTER's evaluated on the 15 seen languages and the rows from "101" to "PL" are PTER's evaluated on the 5 unseen languages. The row AvgS the is the average PTER over the 15 seen languages and the row AvgU are the average PTER over the 5 unseen languages.

Exp	Base	W2v	W2vm	W2vl	W2vlm	W2vg	W2vgm
103	40.2	41.3	41.1	39.0	39.0	**38.2**	**38.2**
107	52.3	36.6	36.6	32.6	32.6	**32.0**	**32.0**
206	42.4	39.0	38.8	35.9	35.9	**35.2**	**35.2**
307	44.7	43.1	43.1	39.1	39.1	**38.0**	**38.0**
402	47.0	48.9	48.4	44.9	44.9	**44.2**	**44.2**
404	**38.0**	42.2	41.7	39.1	39.1	38.6	38.6
N	21.3	15.3	15.3	14.0	14.0	**13.2**	**13.2**
CZ	11.0	10.5	10.5	9.1	9.1	**8.5**	**8.5**
FR	13.7	14.8	14.8	12.9	12.9	**12.1**	**12.1**
CH	30.0	17.2	17.2	15.9	15.9	**15.5**	**15.5**
TH	26.1	22.2	22.2	19.9	19.9	**18.9**	**18.9**
GE	26.1	25.1	25.1	23.2	23.2	**22.3**	**22.3**
PO	18.4	18.7	18.7	16.3	16.3	**16.0**	**16.0**
TU	21.3	21.0	21.0	19.3	19.3	**18.4**	**18.4**
BG	27.0	30.2	30.2	28.2	28.2	**26.9**	**26.9**
101	77.0	77.9	76.5	74.6	**73.1**	76.1	**73.1**
203	78.2	79.3	76.8	76.3	72.8	72.4	**69.3**
CR	47.8	47.3	42.8	41.3	**35.2**	50.8	39.6
SP	38.1	39.0	36.8	37.3	**34.4**	37.5	35.3
PL	62.5	66.7	61.2	59.8	**54.0**	61.9	56.3
AvgS	30.6	28.4	28.3	26.0	26.0	**25.2**	**25.2**
AvgU	60.7	62.0	58.8	57.9	**53.9**	59.7	54.7

namely Cantonese, Lao, Croatian, Spanish, and Polish as the testing language set and use the remaining 15 languages as a training language set. Each testing language is selected to have a similar language belonging to the same language family in the training set.

6.4.2 Data Preprocessing

We use ESPNET as our ASR framework [194] since ESPNET offers a complete ASR pipeline including data preprocessing, Transformer implementation, network training and decoding.

Due to the sampling rate difference between different corpora, we upsample all audio signals to 16kHz. Using KALDI [181], we then extract 80-dim log Mel spectral coefficients with 25ms frame size and 10ms shift between frames, and augment the frame vectors with 3 extra dimensions for pitch features.

The transcripts are converted to IPA symbols using LANGUAGENET [102] G2P models and the unique IPA symbols, including base phones, diacritics and suprasegmentals, in all 15 training languages are collected as the shared phonetic token inventory. The resulting inventory size is 95. The test languages contain phones that are not present in any training languages, which causes an out-of-vocabulary (OOV) problem as our network cannot predict a phone it has never seen. We map each OOV phone to its closest in-vocabulary phone according to its articulatory features defined by IPA. For example, /β/ in Spanish is mapped to /v/.

6.4.3 Language Embedding

We experiment with two types of transformations to generate the language embedding, a 3-layer fully-connected transformation and a 3-layer graph-convolutional transformation[3] on the language representations extracted from GLOTTOLOG [186] and PHOIBLE [188]. Each transformation layer is followed by a ReLU activation and a dropout layer with a dropout rate of 10%. The output of the transformation networks is used as language embedding and as input to the self-attention based ASR network.

[3] `https://github.com/tkipf/gcn`

6.4.4 Model

We experiment with two audio embedding modules. One consists of two 2D convolutional layers (randomly initialized) with a subsampling factor of 4 that takes the extracted 83-dim audio features as input, and the other is the feature extractor of a pretrained Wav2vec2 [11] model that directly takes the 16kHz waveform as input. We fix the weights of the Wav2vec2 feature extractor during training.

The encoder of our model architecture is similar to the transformer architecture in [195]. The audio embeddings are fed into 12 self-attention encoder layers, each having 4 heads, an attention dimension of 256 and a 2048-dim position-wise feed-forward layer. The only difference is that input to each encoder layer is additionally concatenated with the correct language embedding to provide language information to the transformer.

Our preliminary experiments indicate that the self-attention decoder framework does not outperform a simple CTC decoder in cross-lingual recognition, which is consistent with the findings in [191]. Therefore, we discard the self-attention decoder in [195] and apply a dense layer to the encoder output to compute the frame-wise phoneme posteriors and the CTC loss.

6.4.5 Evaluation

We use phonetic token error rate (PTER) [75] to evaluate our models. It is calculated the same way as character error rate except that the model predicts a set of language-universal IPA tokens instead of normal orthographic characters. It treats diacritics (such as aspiration /ʰ/), suprasegmentals (such as long vowels /ː/ and primary stress symbol /ˈ/), and tones (such as high tone /˥/ and low tone /˩/) as separate tokens. It also splits diphthongs and affricates into individual symbols. For example /ˈtaː/ would be viewed as 4 tokens. Therefore, our PTER metric slightly differs from the phone error rate (PER) calculated in other multilingual literature such as [76].

6.5 Results

6.5.1 Multilingual and Cross-lingual Phonetic Recognition

We train and test on our 20-language dataset with 7 different models: BASE, W2V, W2VM, W2VL, W2VLM, W2VG, W2VGM. All the models have a self-attention encoder and a CTC decoder. BASE model uses a randomly initialized 2D convolutional feature extractor and the models with 'W2V' prefix instead use a pretrained WAV2VEC2 feature extractor. The models with 'L' and 'G' suffices have an additional linear or graph-convolutional transformation network to compute the language embeddings. Models with 'M' suffix apply phone inventory masking to the softmax output layer of the decoder.

The performance is shown in Table 6.2, where both proposed models (W2VLM and W2VGM) outperform the BASE model; W2VGM model achieves the lowest multilingual error rate, while W2VLM model achieves lowest cross-lingual error rate.

By comparing BASE and W2V, we see that a pretrained WAV2VEC2 feature extractor reduces the average multilingual recognition error rate. In particular, the reduction is 15.7% on Vietnamese (107), 6% on Dutch (N) and 12.8% on Mandarin (CH). Although it slightly increases the cross-lingual error rate, we decide to build on W2V model instead of BASE model.

Comparing the average test PTER (AVGU) of W2V, W2VL and W2VG with that of W2VM, W2VLM and W2VGM, we see that masking out the non-existing phonetic tokens in the test language greatly improves the recognition accuracy, possibly due to the reduced prediction space. The W2VGM model, which places the most emphasis on language-family structure, gains the largest improvement from phone masking, but still does not outperform the W2VLM model, suggesting that applying the graph constraint a second time (GCN on top of NODE2VEC embeddings) provides no extra reduction of PTER.

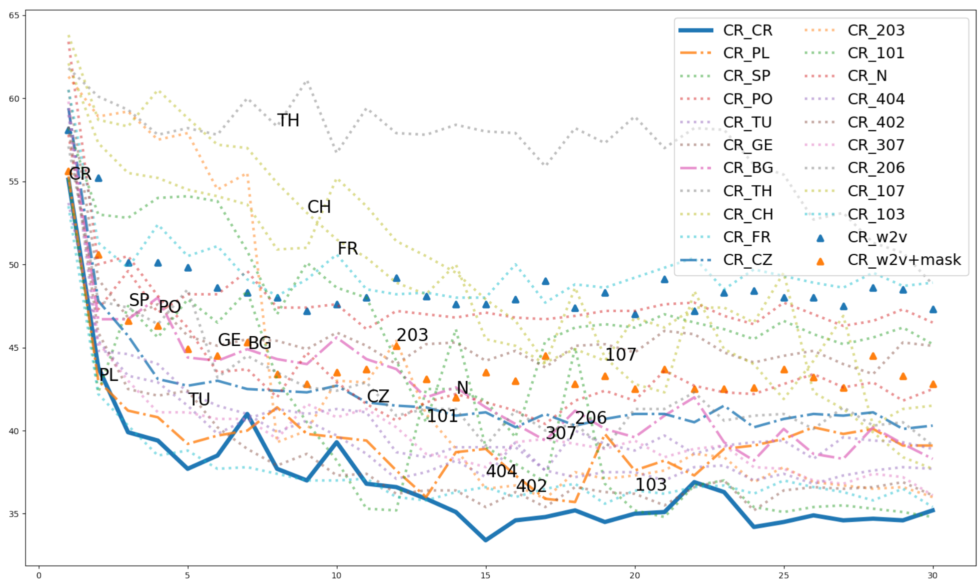


Figure 6.2: PTER of W2vLM model tested on Croatian with correct and fake language labels.

6.5.2 Cross-lingual Phonetic Recognition with Fake Language Labels

To better understand how language embedding affects the model's performance, we feed both true and fake language embeddings to the model and plot the test PTERs across epochs. Figure 6.2 shows the PTER of W2vLM model tested on Croatian. The blue and orange triangle points are PTERs of the W2v and W2vM models respectively. The blue solid line labeled "CR_CR" is the PTER curve with correct Croatian embedding and the dash-dotted lines or dotted lines are PTER's of the model when provided with fake language embeddings.

When provided with the correct language embedding (CR_CR), the model outperforms the masked wav2vec baseline (W2vM). The PTER of the model when provided with fake embedding varies from 35% to 80%. In particular, when provided with fake embeddings of languages from the same language family, Slavic family in this example, the model generally has a lower PTER compared to others, as shown by the curves of Polish (CR_PL), Bulgarian (CR_BG) and Czech (CR_CZ). This indicates that our model is able to leverage the phylogenetic and phonetic similarities for better accuracy.

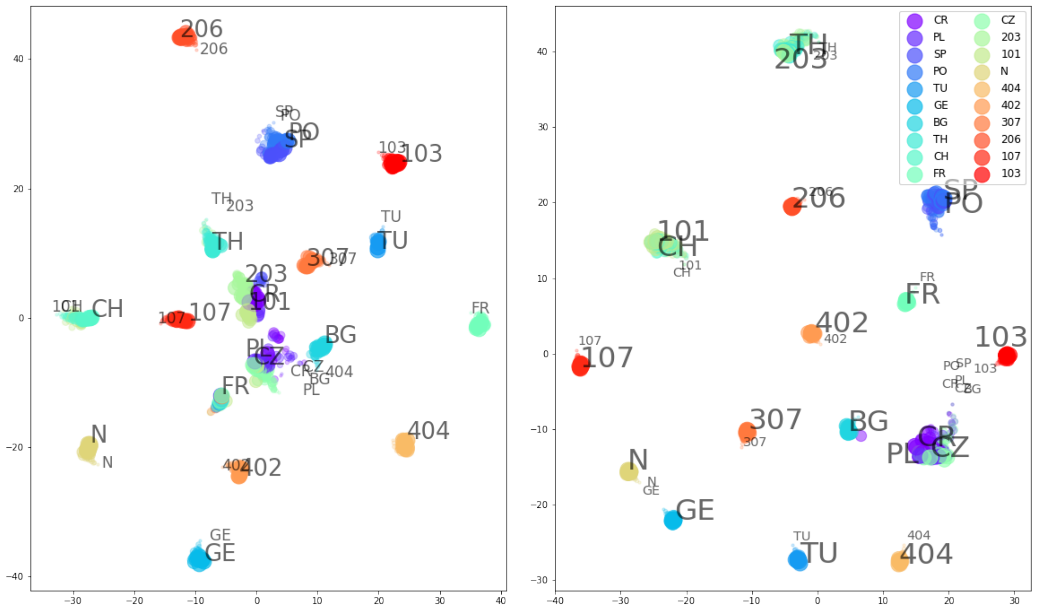


Figure 6.3: t-SNE plot of language embedding. The left plot is the embedding from W2VL and the right plot is the embedding from W2VG.

6.5.3 Visualization of Language Embedding

We visualize the language embeddings of W2VL and W2VG using t-SNE [196] in Figure 6.3. The small and light circles are the embeddings from earlier epochs and the large and solid circles are from later epochs. We use small and light text to label the embeddings' initial-epoch position and large and solid text to label the final-epoch position. In the right plot, we observe that graph convolutional transformation on language vectors largely preserves the phylogenetic information; the languages that are close in the initial epoch remain close in the final epoch. In contrast, the left plot shows that linear transformation preserves the phylogenetic information only partially. For example, while the Sinitic-language embeddings (CH and 101) are close initially, Cantonese (101) moves away from Mandarin (CH) towards the Slavic-languge embeddings (CR, CZ, PL and BG) as the training epoch increases. This observation indicates the linear transformation has larger flexibility to learn its embeddings; as shown in Table 6.2, this flexibility reduces the cross-lingual error rate.

Table 6.3: Phonetic token error rates (PTER) Ablation Study.

W2vlm	101	203	CR	SP	PL	Avg
Glottolog+phoible	73.1	72.8	35.2	**34.4**	54.0	53.9
Glottolog	**69.5**	73.4	**35.1**	34.8	55.7	**53.7**
Phoible	76.0	**71.9**	36.6	38.8	**53.4**	55.3

6.5.4 Ablation Study on Language Representation

We conduct an ablation study to see the role of the Glottolog vector and Phoible vector in error rate reduction by training W2vl model with only Glottolog vector, with only Phoible vector and with both. The results are shown in Table 6.3. First, providing external information reduces error: all three settings (Glottolog, Phoible, Glottolog+Phoible) beat the W2vm baseline. Second, using only Glottolog vectors reduces the Cantonese (101) error rate to 69.5% but raises the Lao (203) error rate to 73.4%, which is close to the performance of the W2vgm model, while using only Phoible vectors does the reverse, raising the Cantonese error rate but reducing the Lao error rate. These results show both vectors improve the performance in different ways; W2vlm finds a good trade-off between relying on phylogenetic information and phonetic information. Finally, we notice that using only Glottolog vectors (Glottolog) has nearly the same performance as both vectors (Glottolog+Phoible). We hypothesize that phoneme masking is functioning as a substitution, reducing the necessity of the Phoible vector.

6.6 Summary

We propose to use external phylogenetic and phonetic knowledge from language typologies to improve the cross-lingual phoneme recognizer. We study the performance of learning language embeddings using a linear transformation network and a graph convolutional network and show that both models outperform the baseline. In particular, we show both phylogenetic and phonetic knowledge are necessary for good cross-lingual accuracy and that a linear transformation network can flexibly leverage both types of information to learn a better phonetic model than a graph convolutional network.

CHAPTER 7

TOWARDS FEW-SHOT SPOKEN LANGUAGE UNDERSTANDING WITH FROZEN LANGUAGE MODELS

7.1 Introduction

Large-scale pretrained language models (PLM) [68, 50, 51, 52] have brought great success in natural language processing (NLP) [197]. Recently, researchers have discovered that PLMs demonstrate a strong capability for few-shot learning on many NLP tasks [52, 198]. Specifically, if we[1] feed to a language model a prefix containing several text-prompt-answer demonstrations of a task, as well as a new question, a language model can generate a decent answer to the new question upon seeing the prefix. Furthermore, by pretraining an image encoder to generate feature vectors that are meaningful to a PLM, the PLM can be given the ability to solve few-shot image understanding tasks [200].

We are therefore interested in whether such few-shot learning capabilities can generalize to spoken language understanding (SLU) tasks as well. More specifically, our setting is as follows. Given a certain task, the task demonstrations are in the form of triplets containing (1) a speech utterance, (2) a text question/prompt, and (3) a text answer. We also have a new question that is in a similar form to the demonstrations but without an answer. Our goal is to convert the task demonstrations and the new question into a text prefix and feed it to a fixed language model, so that it can produce answers to the new question. Figure 7.1(b) shows an example, where the model is being taught to identify the gender of the person discussed in the speech utterance by seeing a few short demonstrations, each containing three components: first, a speech utterance (saying, e.g., ‘a woman in a red suit’), then a text prompt (‘the speaker is describing a’), and finally the text an-

[1] The project described in this chapter is part of [199], with co-authors Junrui Ni, Yang Zhang, Kaizhi Qian, Shiyu Chang, and Mark Hasegawa-Johnson. The code is available in `https://github.com/Hertin/WavPrompt`.

swer ('woman'). Concatenated to the end of the training demonstrations is a question in a similar form but without the answer; the model is judged to perform correctly if it generates the correct answer (e.g., either 'man' or 'woman').

The main challenge of this task is to convert the speech into a form that can be accepted by the language model as the text prefix. One naïve way is simply to convert the speech to text using an automatic speech recognition (ASR) system, and then perform few-shot learning on the transcribed demos the same way as in NLP tasks. However, such a naïve paradigm would propagate the errors in ASR to the language model, thereby undermining its few-shot learning performance. Also, this naïve solution could not handle non-speech audio understanding tasks. We thus ask: are there better end-to-end solutions to speech understanding tasks?

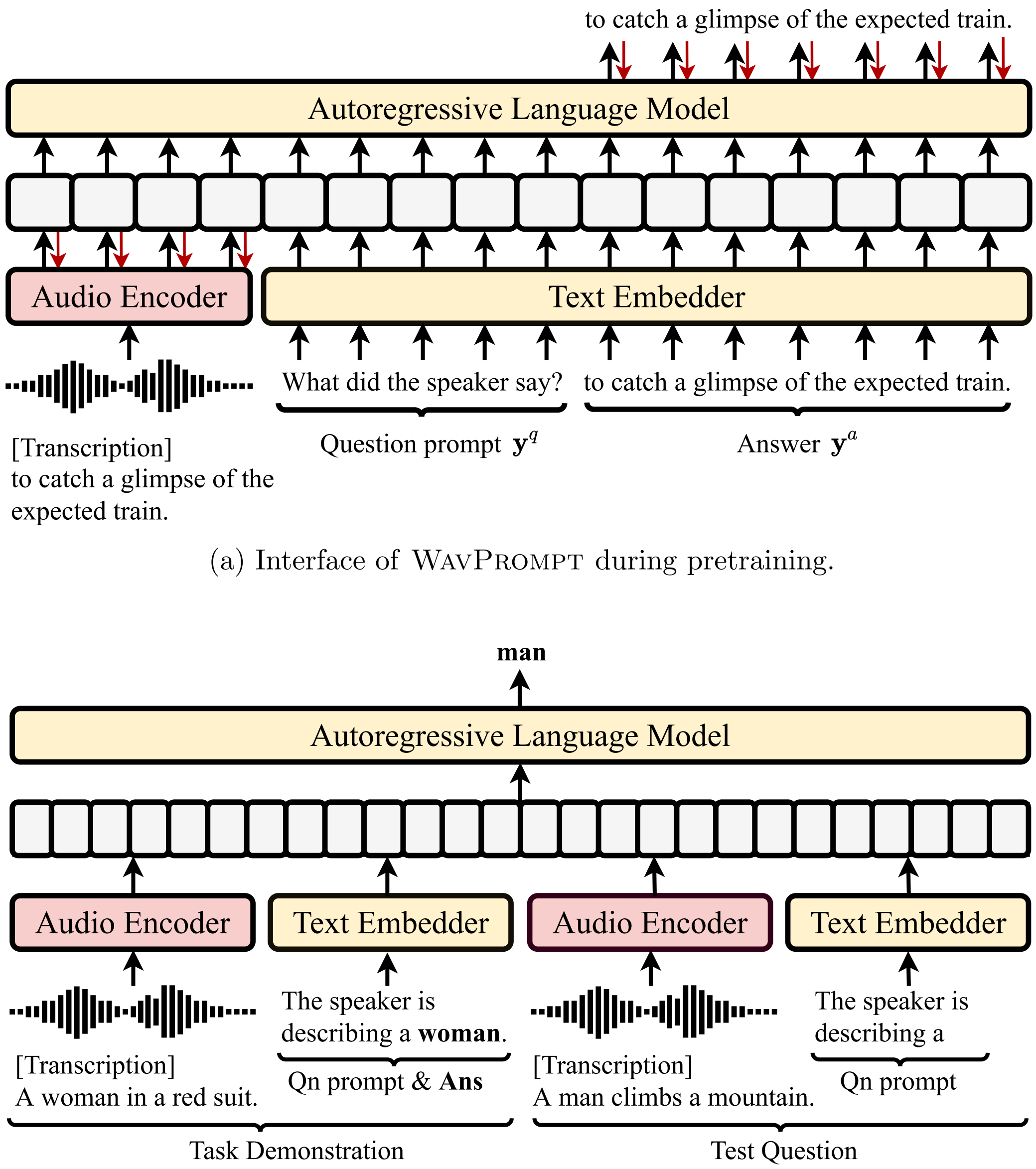
In this paper, we propose WavPrompt, an end-to-end few-shot learning framework for speech or audio understanding tasks. WavPrompt consists of an audio encoder and a language model. The audio encoder is pretrained as part of an ASR, so that it learns to convert the speech in the demonstrations into embeddings digestible to the language model. After pretraining, the entire framework is frozen and ready to perform few-shot learning upon seeing the demonstrations.

We evaluate our model on speech classification tasks, and we can confirm that the zero-shot learning capabilities of fixed language models do generalize to simple speech understanding tasks. Furthermore, WavPrompt, with its end-to-end pipeline, achieves a significant gain over the aforementioned naïve ASR+NLP baseline. We further perform an extensive ablation study in search of the best hyperparameter settings. The findings of this paper can provide guidance and insights for research towards next-generation few-shot learning for speech and audio understanding.

7.2 Methods

7.2.1 Model Architecture

Figure 7.1 shows the architecture of WavPrompt, which consists of a pretrained audio encoder f_ϕ, for which we use the wav2vec 2.0 base model [11],



(a) Interface of WavPrompt during pretraining.

(b) Interface of WavPrompt during inference.

Figure 7.1: Interface of WavPrompt

and a pretrained autoregressive language model, for which we use the GPT2 [51]. The audio encoder f_ϕ encodes the speech audio $\mathbf{x}$ into continuous audio embeddings $\mathbf{s} = [s_1, s_2, ..., s_m] = f_\phi(\mathbf{x})$. The language model contains a text embedder h_θ that converts the text $\mathbf{y} = [y_1, y_2, ..., y_l]$ into a sequence of text embedings $\mathbf{t} = [t_1, t_2, ..., t_n] = h_\theta(\mathbf{y})$ and a transformer-based neural network g_θ that models the text distribution $p(\mathbf{y})$ as

$$\log \mathbb{P}(\mathbf{y}) = \sum_{i=1}^{n} \log \mathbb{P}(t_i | t_1, ..., t_{i-1}) = \sum_{i=1}^{n} g_\theta(t_1, ..., t_{i-1})_{t_i} \tag{7.1}$$

7.2.2 Downsampling Layer

The wav2vec audio encoder takes 16 kHz audios and extracts feature vectors at a frequency of 50 Hz. A simple calculation gives us that the LibriSpeech ASR corpus [73] has an average of 2.7 words per second and 4.9 tokens per second using GPT2's tokenizer. This means the text embedding vectors have a frequency of roughly 5 Hz, which is only 10% the rate of the audio embeddings. Therefore, we append a downsampling layer after the audio encoder to reduce the rate of audio embeddings, so that the rate of the audio embedding can better match that of the text embeddings.

7.2.3 Speech Recognition Pretraining

We pretrain WavPrompt as an ASR, using the 100-hour train-clean split of the LibriSpeech ASR corpus [73]. We also create 5-hour and 10-hour pretraining datasets by sampling from the 100-hour split to simulate low resource conditions.

We keep the language model fixed and only update the audio encoder during pretraining. An overview of the pretraining interface is shown in Figure 7.1(a), where the red arrows denote the back-propagation of the gradients. The audio embeddings $\mathbf{s}$, together with the text embeddings $\mathbf{t}^q = [t_1^q, t_2^q, ..., t_n^q]$ of the question prompt $\mathbf{y}^q$ are fed to the language model so that the language model models the probability of the answer $\mathbf{y}^a$ conditioned on the audio and the question prompt as

$$\log \mathbb{P}(\mathbf{y}^a|\mathbf{x}, \mathbf{y}^q) = \sum_{i=1}^{l} \mathbb{P}(t_i^a|\mathbf{s}, \mathbf{t}^q, t_1^a, ..., , t_{i-1}^a) \tag{7.2}$$

$$= \sum_{i=1}^{l} g_\theta(s_1, ..., s_m, t_1^q, ..., t_n^q, t_1^a, ..., t_{i-1}^a)_{t_i^a} \tag{7.3}$$

We use the question 'what did the speaker say?' as a prompt during pre-training.

7.2.4 Few-Shot Evaluation

We evaluate WavPrompt on few-shot binary classification tasks. During evaluation, we do not update the model parameters. Instead, the model is given a single prompt sequence that contains from 0 to 10 demonstrations of a new task, followed by a question that it must answer using the form specified in the demonstrations. An illustration of the inference interface during evaluation is shown in Figure 7.1(b). As shown, the question is usually a sentence with a gap at the end; WavPrompt must fill the gap based on the content of the audio. Unlike the setting in [201], we restrict each task to a finite output space (either two or nine possible answers), so that the accuracy of the few-shot learner can be meaningfully compared to chance performance.

7.2.5 Calibration

During the evaluation, we find that the performances of our models are not stable. We therefore implement the calibration technique reported by [198] to reduce the bias introduced by the language models. We empirically find the calibration brings improvement to the classification accuracy in most cases.

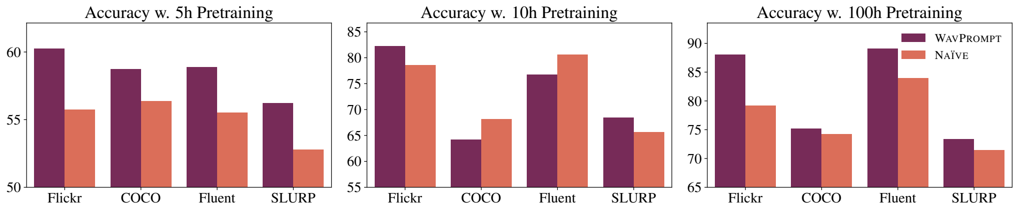


Figure 7.2: Results of speech understanding tasks.

7.3 Experiments

7.3.1 Datasets

We evaluate WavPrompt on four speech datasets: Flickr8k Audio Caption Corpus (Flickr) [202], Fluent Speech Commands Corpus (Fluent) [203], Spoken Language Understanding Resource Package (SLURP) [204] and SpokenCOCO Audio Caption Corpus (SpokenCOCO) [205]. In addtion, we evaluate WavPrompt on a non-speech dataset: Environmental Sound Classification (ESC50) [206]. Brief introductions and the preprocessing steps of the dataset are as follows.

- The Flickr dataset contains 40,000 spoken captions of 8,000 natural images. We drop the images and only use the spoken caption audios and their transcripts. We then randomly sample 2000 captions and manually assign four sets of labels to the captions. We form man-woman label set by assigning 'man' and 'woman' labels to the captions that contain either only 'man' or only 'woman' words respectively. We create the male-female label set by replacing 'man' and 'woman' labels with 'male' and 'female' labels. We repeat the procedure to form black-white and dark-light label sets, but we additionally drop those that are not describing the color of the clothes. The resulting subset contains around 400 man-woman and male-female labeled samples and around 70 black-white and dark-light labeled samples. We use 'the speaker is describing a person in' as the question prompt for the color labels and 'The speaker is describing a' for the gender labels. This dataset is mainly used to probe if the model can capture semantic relations between word-pairs.

- The SpokenCOCO dataset contains approximately $600,000$ spoken captions describing the images in the Microsoft COCO (MSCOCO) dataset [207]. The MSCOCO dataset classifies the images using 12 super-category labels, which we use as the labels of the spoken captions. During evaluation, we ask the model to discern between the 'vehicle' labels and the rest of the labels, forming a total of 11 classification tasks. We use 'The speaker is describing' as the question prompt.

- The Fluent dataset contains spoken commands that interact with smart devices, such as 'play the song' and 'increase the volume.' Each command is labeled with action, object and location. We define topic labels to be the same as the object label most of the time, except that when the action is 'change language,' the topic is set to 'language' instead of the actual language name. We use 'The topic is' as the question prompt.

- The SLURP dataset is an SLU dataset that contains human interaction with home assistants from 18 different domains. We select five domains: 'music', 'weather', 'news', 'email' and 'play' and form ten domain pairs for our model to perform binary classification. We use 'This is a scenario of' as the question prompt.

- The ESC50 dataset contains $2,000$ environmental sounds including animal sounds, human non-speech sounds, natural soundscapes, domestic and urban noises, etc. We use the sound label as groundtruth text and pretrain additional WavPrompt models on the 100-hour Librispeech dataset and ESC50 dataset for ASR and environment sound classification tasks simultaneously. During pretraining, we prompt the model with 'What did the speaker say?' for the ASR task and 'What sound is this?' for the environment sound classification task. We test the model on a subset of the training set that only contains sounds of nine animals: dog, cat, bird, sheep, cow, pig, rooster, hen and frog. During testing we assign a distinct verb to each of the nine animals: barks, chirps, bleats, meows, moos, snorts, crows, clucks, and croaks. WavPrompt needs to predict the correct verb given the animal sound and a few examples. We use '=>' as the question prompt during evaluation.

7.3.2 Experiment Setup and Baseline

We modify the fairseq [150] training pipeline for our experiment. We use the Wav2vec2 base model implemented in fairseq as the udio encoder and the GPT2 model with 117 million parameters implemented in Huggingface [208] as our language model.

For the speech classification tasks, we pretrain a total of 15 WavPrompt models with five downsampling rates (2, 4, 8, 16, 32) under three resource conditions (5, 10 and 100 hours of LibriSpeech data). For the non-speech classification tasks, we pretrain five WavPrompt models with five downsampling rates (2, 4, 8, 16, 32) using 100 hours of LibriSpeech data.

During evaluation, we randomly sample several samples along with their correct labels from the test set as shots. The shots are converted to embeddings and are prepended to the question prompt embeddings. We sample 250 samples from the rest of the test set to form an evaluation batch and drop samples from the class containing more samples to evenly balance the class labels in the batch. As a result, a binary classification accuracy greater than 50% is better than chance. We sample five batches with different random seeds. The classification accuracy we report is the average accuracy over the five batches.

We compare the WavPrompt with the Naïve baseline mentioned in Section 7.1, which converts the speech into text and performs few-shot learning using the transcribed text. Specifically, Naïve uses the same model as WavPrompt. It performs few-shot learning via two steps. First, the speech is converted into text using an ASR. To achieve this, we use the pretrained WavPrompt as an ASR by prompting the language model with the audio embedding and the pretraining question ‘what did the speaker say?’. Second, to perform few-shot learning, we prompt the language model with the transcribed text embeddings instead of audio embeddings. In other words, the only difference between WavPrompt and Naïve is that the audio embeddings are used in the prompt in the former, whereas the transcribed text embeddings are used in the latter.

7.3.3 Results on the Speech Understanding Tasks

Figure 7.2 shows the results on the four speech understanding tasks. To factor out the influence of numbers of shots, we use the best accuracy achieved over all numbers of shots to represent the model's performance on individual pairs of labels, for both WavPrompt and Naïve. We average the accuracy over all label pairs in a dataset as the overall accuracy. We select the best-calibrated model among all the downsampling rates for both WavPrompt and the Naïve to make a fair comparison. We compute the overall accuracy of the model across four speech understanding datasets under three resource conditions.

We observe that both algorithms can achieve an accuracy significantly above chance, which confirms that language models can perform zero-shot learning on speech understanding tasks. Also, the performance increases as the pretraining dataset size increases. Finally, WavPrompt consistently outperforms Naïve in almost all cases across datasets and across resource conditions, which verifies the advantage of the end-to-end framework. We note the 100-hour Naïve achieves a word error rate of 9.07% on Librispeech, but only 44.42% on the four test datasets due to domain mismatch, which may explain its worse performance relative to WavPrompt.

7.3.4 Downsampling Rate

We use the best accuracy over all numbers of shots as the model performance as in Sec 7.3.3. We average the best accuracy over all pairs of labels in each dataset and present the results in Table 7.1. The results are consistent across datasets, suggesting that a downsampling rate of 8 gives the best accuracy when the model is pretrained using 10 or more hours of data and a downsampling rate of 4 gives better accuracy when the model is trained using 5 hours of data. The best downsampling rate being 8 is expected, as it produces the audio embeddings at a rate closest to that of the text embeddings as discussed in Sec 7.2.2.

Table 7.1: Classification accuracy across downsampling rates.

	Dataset	2	4	8	16	32
5h	Flickr	57.82	57.99	**60.27**	56.63	52.28
	COCO	55.08	**58.74**	56.92	56.62	53.79
	Fluent	55.54	**58.88**	57.4	57.35	53.32
	SLURP	54.40	**56.23**	53.94	54.85	51.97
10h	Flickr	64.91	65.77	**82.21**	79.23	56.84
	COCO	59.56	59.18	**64.24**	54.99	55.84
	Fluent	64.21	72.01	**79.80**	64.58	53.41
	SLURP	55.95	57.00	**68.47**	60.53	54.16
100h	Flickr	82.61	**88.03**	85.73	79.96	79.40
	COCO	68.52	67.68	**75.15**	67.77	65.72
	Fluent	82.47	87.36	**89.11**	81.12	83.90
	SLURP	72.05	72.07	**73.37**	68.69	68.63

7.3.5 Calibration

We compare the accuracy with calibration versus without calibration using the best downsampling rate obtained in Table 7.1. For each dataset we average the best classification accuracy over all label pairs for both the model with calibration and without calibration. The results are presented in Table 7.2. Almost in every case the model with calibration outperforms that without calibration by a large margin, suggesting necessity of the calibrating the PLM. The only exception occurs in the model pretrained using 10 hours of data and tested on Flickr dataset, but even in that case the accuracies are comparable.

Table 7.2: Classification accuracy between the model with calibration and without calibration denoted by Cali and NCali respectively.

	5h		10h		100h	
	Cali	NCali	Cali	NCali	Cali	NCali
Flickr	**60.27**	54.89	82.21	**82.42**	**88.03**	84.78
COCO	**58.74**	51.72	**64.24**	59.15	**75.15**	68.95
Fluent	**58.88**	58.50	**79.80**	66.85	**89.11**	87.54
SLURP	**56.23**	55.09	**68.47**	66.04	**73.37**	69.52

7.3.6 Number of Shots

To study the effect of the number of shots, we plot the classification accuracy across different datasets under 100-hour LibriSpeech dataset in the left subplot of Figure 7.3 and plot the accuracy across different resource conditions on Fluent dataset in the right subplot of Figure 7.3. Although the accuracy curves exhibit different patterns across different datasets and different resource conditions, we observe that there usually exist two peaks: one with zero demonstration examples and one with four to six demonstrations. In Flickr and COCO experiments, zero-shot gives the best performance and increasing numbers of shots does not bring any benefits. One possible explanation is that the Flickr and COCO datasets are simpler than the Fluent and SLURP datasets, in the sense that the class labels or their near-synonyms occur directly in the speech; since the model has been pretrained as an ASR, the neurosymbolic representations of these answers may be already activated in the language model, so that the extra activation provided by the question is sufficient to generate a correct answer, even with zero demonstration examples. In Fluent and SLURP experiments, increasing shots to four or six yields the best accuracy but further increasing shots downgrades the performance. Using a larger language model might result in a more consistent pattern, which we leave as future work to explore.

7.3.7 Generalizing to Non-Speech Tasks

We additionally conduct a classification experiment using ESC50, a non-speech dataset. Prompted with a few examples, WavPrompt needs to predict the correct verb corresponding to the animal that makes the non-speech sound. We also provide a text baseline that replaces audio embedding with the text embedding of the animal's name. As in previous sessions, we use the best accuracy across number of shots to represent the model's performance, for both WavPrompt and the baseline. The results are presented in the Table 7.3.

We observe that the classification accuracies are all better than chance, which is 11.11% for a nine-way classification, and the best WavPrompt with a downsampling rate of 8 is slightly better than the text baseline. These results show that WavPrompt is able to extract information from non-

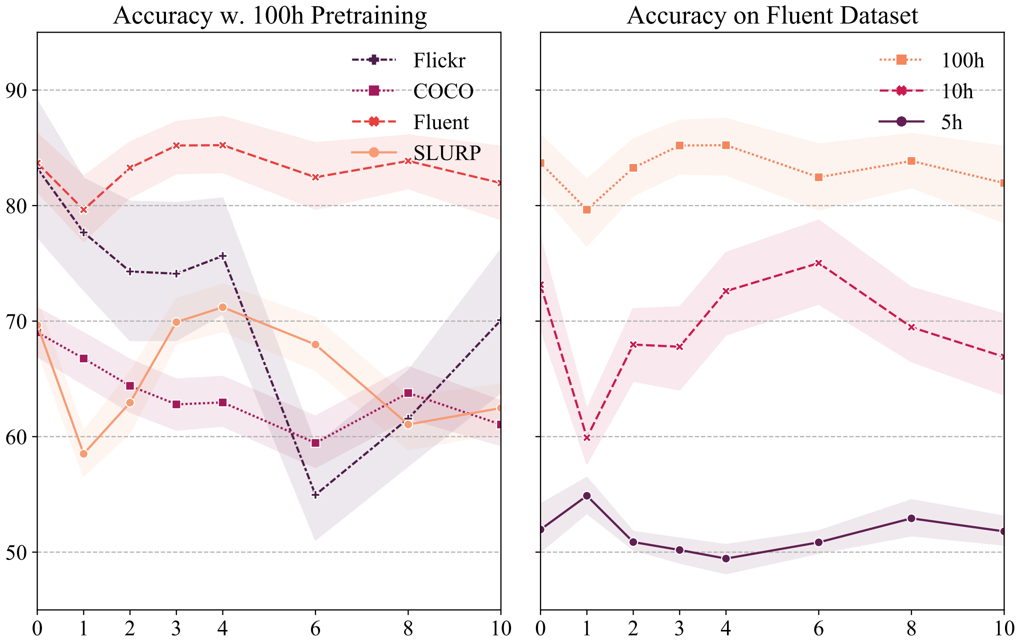


Figure 7.3: Classification accuracy versus number of shots. Shaded region is ±1 standard deviations.

speech audio and then leverage commonsense knowledge from its pretrained language model to solve problems.

Table 7.3: Classification accuracy across downsampling rates on ESC50 dataset.

	2	4	8	16	32	TEXT
ESC50	38.11	31.04	**43.50**	32.89	24.26	42.22

7.4 Summary

In this project, we propose a novel speech understanding framework, WavPrompt, and show that WavPrompt is a few-shot learner that can perform both speech and non-speech understanding tasks better than a naïve ASR baseline. We conduct detailed ablation studies on different components and hyperparameters to empirically identify the best model configuration.

7.5 Acknowledgements

The project in this chapter is supported by the IBM-UIUC Center for Cognitive Computing Systems Research (C3SR).

CHAPTER 8

GRAPHEME-TO-PHONEME TRANSDUCER WITH ACOUSTIC INFORMATION

8.1 Introduction

Grapheme-to-phoneme (G2P) transducers have been extensively applied to generate phoneme transcripts from grapheme transcripts to train phoneme-based automatic speech recognition (ASR) and text-to-speech (TTS) systems. Since grapheme transcripts are the texts in many speech-text corpora, in this paper, we[1] use the terms "text" and "grapheme transcripts" interchangeably while distinguishing them from "phoneme transcripts." Compared to grapheme transcripts, phoneme transcripts have two advantages. First, they directly specify the pronunciation of the utterance, making them a better candidate for distribution matching between the speech and the transcripts. This is supported by the findings in recent unsupervised speech recognition [2] and unsupervised Text-to-Speech synthesis [3] works where lower error rates are reported using phoneme transcripts converted from grapheme transcripts than using grapheme transcripts directly. Second, writing systems vary across different languages. Multilingual training of an end-to-end neural ASR reduces word error rates on languages in the training set [82], but it is not possible to apply such a system to a previously-unseen test language with previously-unseen characters, making it difficult to transfer knowledge across different grapheme transcripts. Phoneme transcripts, such as the International Phonetic Alphabet (IPA), which depicts speech pronunciation, can have a universal annotation for different languages and are, therefore, more suitable for multilingual or cross-lingual transfer learning [209, 77]. A multilingual ASR trained using both grapheme and phoneme transcripts has lower word error rate in under-resourced languages, apparently because it is

[1] The project described in this chapter is part of the submission accepted by ICASSP 2024, with co-authors Mark Hasegawa-Johnson and Chang D. Yoo. The code is available in `https://github.com/Hertin/g2pu`

better able to generalize acoustic knowledge across writing systems [83].

As suggested by the acronym, G2Ps generate phonemes from graphemes, sometimes with the benefit of auxiliary input tags marking part of speech or word sense [103]. A G2P may generate several candidate phone transcripts, from which a forced aligner selects the one that best matches the waveform, and such phone transcripts have been used to re-train the acoustic model [210] and/or the G2P [211], but our literature search has not discovered any system that simultaneously optimizes the acoustic model, the G2P, and the phoneme transcript of the training corpus.

We, therefore, propose our model G2PU (grapheme-to-phoneme transducer with speech units), which consists of a G2P sub-network to generate phoneme transcripts from texts and a U2P sub-network to generate phoneme transcripts from speech units.

We make three assumptions about the training resource: there exists a sufficient amount of non-parallel speech and text data, a limited amount of parallel speech-text data, and a teacher G2P tool. Sufficient non-parallel speech and text data are used to train large self-supervised models to generate speech and text representations that boost the downstream G2P and U2P performance. The limited 100-hour parallel speech-text data are used to train the G2PU that matches the distribution between speech, graphemes, and phonemes. A teacher G2P tool generates coarse phoneme transcripts to complete the parallel speech-text-phoneme triplet, which is the training data for G2PU. The assumption of a teacher G2P is a reasonable compromise to the sparsity of high-quality phoneme transcripts, given a minimal G2P can be constructed from a small lexicon and descriptions of the pronunciation rules of the language [102].

To measure the performance of the proposed G2PU model, groundtruth phoneme transcripts are needed. We choose Chinese and Japanese for our experiments as there are corpora available that contain parallel speech-text-phoneme data. The experiments on these two languages demonstrate that the G2PU model, with a proper weight between the G2P sub-network and the U2P sub-network, can produce better phoneme transcripts than its teacher. Specifically, our model reduces the phoneme error rate (PER) by 7% to 29% relative compared to its teacher.

8.2 Related Works

8.2.1 Existing Grapheme-to-Phoneme Transducers

The resource-rich languages such as English and Mandarin Chinese have many deep neural network based G2P tools such as DeepPhonemizer[2] [59, 212], T5G2P [64], and SoundChoice [67] for English and G2PM[3] [213] and G2PW[4] [103] for Mandarin Chinese. While having low phonetic error rate of the generated phoneme transcripts, these deep G2P tools require access to large corpora of text-phoneme pairs and additional linguistic annotations such as part-of-speech (POS) tags. One example of such a model is SoundChoice [67], which uses a combination of three large datasets including LibriSpeech-Alignments [203], Google Wikipedia Homograph Data [214] and CMUDICT[5] to train a recurrent-neural-network-based G2P. Another model, G2PW [103] finetunes the large pretrained BERT model [68] on Chinese Polyphones with Pinyin (CPP) dataset [215] that contains around $99,000$ sentences augmented with additional POS label extracted using an external tagging tool.

Low-cost G2P models such as LanguageNet can be created for a large number of low-resource languages by training a finite-state transducer (FST) on a small pronunciation lexicon and/or a table of letter-to-sound rules. However, these models suffer from varied PER ranging from 7% to 45% [102]. To improve G2P models, additional linguistic knowledge can be incorporated such as phrase segmentation and hand-crafted rules [216].

According to the case study comparing the performance of the rule-based FST LanguageNet and the neural network based G2PW in Sec 3.2.3, the performance of the G2P tools can largely affect the performance of the ASR models trained on the transcripts they produce. In order to better study the difference between the transcripts generated by the two G2P tools, We convert the *pinyin* transcript to IPA transcript using DragonMapper[6]. Using the DragonMapper IPA transcripts as a reference, we compute the phoneme error rate (PER) of the LanguageNet IPA transcript, and the re-

[2] https://github.com/as-ideas/DeepPhonemizer

[3] https://github.com/kakaobrain/g2pM

[4] https://github.com/GitYCC/g2pW

[5] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[6] https://github.com/tsroten/dragonmapper

Table 8.1: PER of LanguageNet IPA transcripts compared against G2PW IPA transcripts.

G2P	PER
LanguageNet	38.35
+Move Tone	24.25
+Map Diphthong	18.14
+Map Phone	11.98
+Remove Tone	5.51

sults are summarized in Table 8.1 which is 38.35%. we notice there are subtle differences between the two IPA transcripts such as the places of tone annotation and the phoneme inventory. For a fairer comparison, we normalized the LanguageNet IPA transcripts by moving the tone annotations after the nasals /n/ and /ŋ/, and mapping the diphthongs and phones to the ones used in DragonMapper IPA transcripts. The normalization reduces the PER to 11.98%. After we remove tones in both IPA transcripts, the PER is further reduced to 5.51%, which suggests half of the mistakes are tone mistakes and half are phone mistakes. This case study indicates that there is a large quality gap between the minimal rule-based LanguageNet and the deep neural network based G2PW that is trained on large G2P corpora and incorporates additional linguistic knowledge.

Although incorporating prior linguistic knowledge can improve G2P models, it can be challenging to apply this strategy to low-resource languages due to the diversity of linguistic structures across languages. For example, the G2PW model assumes limited phoneme combinations and a one-to-one correspondence between characters and phoneme combinations, which may not hold true for languages like English and Japanese. Consequently, the G2P problem cannot be reduced to a simple classification task as in Chinese, making it difficult to apply similar strategies to these languages.

8.2.2 Forced Aligner as Grapheme-to-Phoneme Transducer

Forced alignment [181, 210] iteratively trains acoustic models to estimate the likelihood of a phone given its acoustic features and aligns the most probable phoneme from the lexicon to the corresponding speech frames. It can be viewed as a G2P tool relying on additional acoustic information from speech.

8.2.3 G2P Transducer with Pretrained Language Models

Self-supervised large pretrained language models have advanced state-of-the-art performance in various downstream tasks in natural language processing (NLP) [68, 69] and ASR [11, 12, 70]. G2P tools benefit from the pretrained text models [217, 67, 215, 218, 103] which encode text tokens into high-quality contextual embeddings. Recently, there have been works on textless NLP that train language models [155] and synthesize speech [168, 70] using discrete acoustic units rather than texts. The units are obtained by performing K-Means clustering on the speech representations extracted from self-supervised models such as Wav2vec2 [11] and HuBERT [12] and the cluster indices serve as the speech units. Although these units are discretized, the audios synthesized from them are of decent quality, indicating that the acoustic units preserve the pronunciation information in the utterances.

8.3 Problem Formulation

Suppose we have a corpus of parallel speech-text samples $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i, \mathbf{z}^i)\}_{i=1}^N$ where N is the total number of speech samples. Let $\mathbf{x}^i = [x_1^i, x_2^i, \dots, x_{T_i}^i]$ denote the i^{th} sample of speech, $\mathbf{y}^i = [y_1^i, y_2^i, \dots, y_{S_i}^i]$ denote its corresponding text transcripts and $\mathbf{z}^i = [z_1^i, z_2^i, \dots, z_{R_i}^i]$ denote its corresponding phoneme transcripts, where T_i, S_i and R_i are the lengths of the speech sample, the length of the text transcripts and the length of the phoneme transcripts respectively. Typically, T is far greater than S and R: $T > S$ and $T > R$. We would like to model $\mathbb{P}_{Z|X,Y}(\mathbf{z}^i|\mathbf{x}^i, \mathbf{y}^i)$, the probability of the phoneme transcripts given the text and the speech sample.

Assume we have a grapheme-to-phoneme transducer G2P that models $\mathbb{P}_{Z|Y}(\mathbf{z}|\mathbf{y})$, the probability of the phoneme transcripts given the text and a unit-to-phoneme transducer U2P that models $\mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x})$, the probability of the phoneme transcripts given the speech. In the ideal case, the text and the speech are two observations of the same phoneme sequence, and either the text or the speech alone is sufficient to determine the correct phoneme transcription. In other words, the phoneme transcript and the text are conditionally independent given the speech and likewise, the phoneme transcript and the speech are conditionally independent given the text, or:

$$\mathbb{P}_{Z|X,Y}(\mathbf{z}|\mathbf{x},\mathbf{y}) = \mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x}), \tag{8.1}$$

$$\mathbb{P}_{Z|X,Y}(\mathbf{z}|\mathbf{x},\mathbf{y}) = \mathbb{P}_{Z|Y}(\mathbf{z}|\mathbf{y}). \tag{8.2}$$

We can combine the prediction from the U2P model and from the G2P model to obtain a more robust prediction by making a weighted geometric average over the probability of the phoneme transcripts predicted by the two models, or:

$$\mathbb{P}_{Z|X,Y}(\mathbf{z}|\mathbf{x},\mathbf{y}) \simeq \mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x})^{\lambda}\mathbb{P}_{Z|Y}(\mathbf{z}|\mathbf{y})^{1-\lambda}, \tag{8.3}$$

where λ is a tune-able hyperparameter.

The G2P and the U2P model are two sequence-to-sequence (S2S) neural network transducers. Given the training set of text-speech-phoneme triplets, the loss function $\mathcal{L}$ can be written as:

$$\begin{aligned}\mathcal{L} &= -\log \mathbb{P}_{Z|X,Y}(\mathbf{z}|\mathbf{x},\mathbf{y}) \\ &\simeq -\lambda \log \mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x}) - (1-\lambda)\log \mathbb{P}_{Z|Y}(\mathbf{z}|\mathbf{y}) \\ &= \lambda \mathcal{L}_{\text{U2P}} + (1-\lambda)\mathcal{L}_{\text{G2P}}.\end{aligned} \tag{8.4}$$

8.3.1 Connectionist Temporal Classification (CTC)

There are two popular neural architectures for S2S transducers. One is based on the connectionist temporal classification (CTC) [4]. Denote the spectral features of the i^{th} utterance as a set of frames $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_{T_i}^i]$ where T_i is the number of speech frames. Denote the reference phoneme transcription as $\mathbf{z}^i = [z_1^i, z_2^i, \ldots, z_{R_i}^i] \in \mathcal{Z}^+$, and the ASR output hypothesis as $\hat{z}^i = [\hat{z}_1^i, \hat{z}_2^i, \ldots, \hat{z}_{\hat{R}_i}^i] \in \mathcal{Z}^+$, where R_i and $\hat{R}_i$ are the lengths of the reference and hypothesis transcriptions of i^{th} sample and $\mathcal{Z}$ is the set of all transcription characters. The true conditional probability distribution $\mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x})$ is unknown; the ASR computes an estimated distribution $\mathbb{P}_{\hat{Z}|X}(\mathbf{z}|\mathbf{x})$ in order to minimize the cross-entropy of the training corpus,

$$\mathcal{L}_{CTC} = -\sum_{i=1}^{|\mathcal{D}|} \log \mathbb{P}_{\hat{Z}|X}(\mathbf{z}^i|\mathbf{x}^i), \tag{8.5}$$

where $\mathcal{D}$ is the training corpus containing utterances with known transcriptions.

CTC [4] performs time-scale modification by positing an alignment sequence, $\Pi^i = [\Pi^i_1, \ldots, \Pi^i_{T_i}]$ whose instance value is $\pi^i = [\pi^i_1, \ldots, \pi^i_T]$. Each time-aligned character π^i_t is either one of the transcription characters ($\pi_t = z_r$ for some r), or $\pi_t = \varnothing$ where $\varnothing$ is a special "blank" character. For example, suppose we have a 5-character text "hello" ($R = 5$) encoded in a 14-frame speech waveform ($T = 14$); the transcription and alignment might be

$$\mathbf{z} = [h, e, l, l, o], \tag{8.6}$$

$$\pi = [h, h, e, e, e, \varnothing, \varnothing, l, l, l, \varnothing, l, \varnothing, o]. \tag{8.7}$$

Training data are often provided with only the transcriptions, and the alignment information is not given. If the alignments are known, it would be easier to estimate the CTC loss $\mathcal{L}_{CTC}$ given in Eq. (8.5) by taking the sum of the log-probabilities of the correct alignment at each frame.

Since alignment is not known, CTC computes the cross-entropy by marginalizing over all the possible alignments that can be mapped to the true transcription using a surjective time-compression function defined as:

$$\mathcal{B} : (\mathcal{Z} \cup \{\varnothing\})^+ \rightarrow \mathcal{Z}^+. \tag{8.8}$$

A commonly used $\mathcal{B}$ first removes repeated labels and then removes all "blank" characters. For any valid alignment π, $\mathcal{B}(\pi)$ is a unique y. For any valid y , $\mathcal{B}^{-1}(y)$ is the set $\{\pi : \mathcal{B}(\pi) = y\}$. The negative log-probability of a transcription $\mathbf{z}^i$ given the input frames x^i can therefore be computed as

$$\begin{aligned}\mathcal{L}^i_{CTC} &= -\log \mathbb{P}_{\hat{Z}|X}(\mathbf{z}^i|\mathbf{x}^i), &(8.9)\\ &= -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{z})} \prod_{t=1}^{T} \exp(e_t(\pi_t)), &(8.10)\\ &= -\operatorname*{logsumexp}_{\pi \in \mathcal{B}^{-1}(\mathbf{z})} \sum_{t=1}^{T} e_t(\pi_t). &(8.11)\end{aligned}$$

where $e_t(\pi_t)$ is the log output of a softmax layer predicting the transcription label at time t. The input of this softmax layer can be a bidirectional LSTM, Transformer, or other neural network parameterized by Θ and having access to the whole sequence $\mathbf{x}$.

Eq. 8.11 requires the enumeration of all possible alignments that produce the transcript $\mathbf{z}$, which is exponential over the input length using vanilla approaches. Graves *et al.*, 2006 [4] show that it can be computed efficiently using the forward-backward algorithm as

$$\mathbb{P}_{\hat{Z}|X}(\mathbf{z}|\mathbf{x}) = \sum_{u=1}^{|\mathbf{z}|} \frac{\alpha_t(r)\beta_t(r)}{e_t(z_r)}, \tag{8.12}$$

where $\alpha_t(r)$ is the forward variable, representing the total probability of all possible alignments for the prefixes $z_{1:r}$ that end with the r^{th} label, and $\beta_t(u)$ is the backward variable of all possible alignments for the suffixes $(z_{r:R})$ that start with the r^{th} label. The network can then be trained with the back-propagation by taking the derivative of the loss function with respect to $e_t(k)$ for the label k.

The training loss is, therefore, the summation over CTC loss of each speech sample as

$$\mathcal{L}_{CTC} = \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}^i_{CTC}. \tag{8.13}$$

8.3.2 Attention-Based Encoder-Decoder Transducer

Unlike the CTC-based architecture, the attention-based encoder-decoder transducer does not make the assumption that each frame corresponds to a phoneme label and that the phoneme labels are conditionally independent given their frame features. Instead, it directly models the posterior probability $\mathbb{P}_{Z|X}(\mathbf{z}^i|\mathbf{x}^i)$. As in the attention-based transducer framework [7], the log-probability is estimated by modeling each character output y_s^i as a conditional distribution given the previous characters $y_{1:s-1}^i$ and the input signal x^i. Using the chain rule, the loss $\mathcal{L}_{att}^i$, i.e., the negative log-probability is computed as

$$\begin{aligned}\mathcal{L}_{ATT}^i &= -\log \mathbb{P}_{\hat{Z}|X}(\mathbf{z}^i|\mathbf{x}^i), && (8.14)\\ &= -\sum_{r=1}^{R_i} \log \mathbb{P}(z_r^i|\mathbf{z}_{1:r-1}^i, \mathbf{x}^i). && (8.15)\end{aligned}$$

The training loss is, therefore, the summation of the attention loss of each speech sample as

$$\mathcal{L}_{ATT} = \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}_{ATT}^i. \tag{8.16}$$

8.3.3 Joint CTC-Attention Learning and Decoding

Joint CTC-attention framework trains an encoder-decoder model using the attention loss $\mathcal{L}_{ATT}$ and additionally trains the encoder using the CTC loss $\mathcal{L}_{CTC}$. This approach has been shown to improve the CER and WER of the deep neural ASR system [219]. During inference, the joint CTC-attention beam search [220] is applied to find the most probable phoneme transcript $\hat{\mathbf{z}}$ given the speech $\mathbf{x}$:

$$\begin{aligned}\hat{\mathbf{z}} &= \operatorname*{argmax}_{\mathbf{z}'\in\mathcal{Z}^*} \log \mathbb{P}_{\hat{Z}|X}(\mathbf{z}'|\mathbf{x}), && (8.17)\\ &= \operatorname*{argmax}_{\mathbf{z}'\in\mathcal{Z}^*} [\lambda \log \mathbb{P}_{CTC}(\mathbf{z}'|\mathbf{x}) + (1-\lambda)\log \mathbb{P}_{ATT}(\mathbf{z}'|\mathbf{x})], && (8.18)\end{aligned}$$

where $\mathbb{P}_{CTC}(\mathbf{z}'|\mathbf{x})$ is the posterior probability $\mathbb{P}_{\hat{Z}|X}(\hat{\mathbf{z}}|\mathbf{x})$ estimated by the encoder-CTC model and $\mathbb{P}_{ATT}(\mathbf{z}'|\mathbf{x})$ is the posterior probability estimated by the encoder-decoder model.

Following Kim *et al.*, 2017 [219] and Hori *et al.*, 2017 [220], we use an encoder-decoder architecture as the G2P transducer to model $\mathbb{P}_{Z|Y}(\mathbf{z}|\mathbf{y})$, the posterior probability of phoneme transcripts given the texts and use an encoder-CTC architecture as the U2P transducer to model $\mathbb{P}_{Z|X}(\mathbf{z}|\mathbf{x})$, the posterior probability of phoneme transcripts given the speech. The reason to use encoder-decoder architecture for the G2P transducer is that this architecture does not make assumptions about the linguistic structure of the text and does not require the input length to be longer than the output length. During inference, we adapt the one-pass decoding strategy [220] where the hypotheses are scored using, $g_{CTC}(h, X)$ and $g_{ATT}(h, Y)$, the scores predicted by the G2P and the U2P model respectively. The scores are computed as

$$\begin{aligned} g_{CTC}(h, X) &= \log \mathbb{P}_{CTC}(h, \dots |X) && (8.19)\\ &= \log \sum_{\nu \in (\mathcal{Z} \cup \langle eos \rangle)^+} \mathbb{P}_{CTC}(h \cdot \nu | X) && (8.20)\\ &= \log \sum_{t=1}^{T} \alpha_{t-1}(|h| - 1) e_t(h_{|h|}) && (8.21)\\ g_{ATT}(h, Y) &= \log \mathbb{P}_{ATT}(h, \dots |Y) && (8.22)\\ &= \sum_{r=1}^{|h|} \log \mathbb{P}_{ATT}(h_r | h_{1:r-1}, Y), && (8.23) \end{aligned}$$

where h is the running hypothesis. Eq. 8.21 follows the definition by Seki *et al.* [221].

8.4 Architecture

To incorporate both linguistic and speech information into the G2P conversion, the G2PU model consists of three parts, a transformer-based encoder-decoder grapheme-to-phoneme (G2P) network, an encoder-CTC unit-to-phoneme (U2P) network and an optional grapheme-to-unit (G2U) network for cases where parallel speech is missing in the test set. The overview of the architecture is shown in Figure 8.1. Both the G2P and the U2P networks are trained using phoneme transcripts generated from a teacher G2P tool, which can be a forced aligner or a rule-based lexicon lookup table. We use the hybrid beamsearcher [219] implemented in ESPNet [194] to jointly decode the G2P and U2P networks.

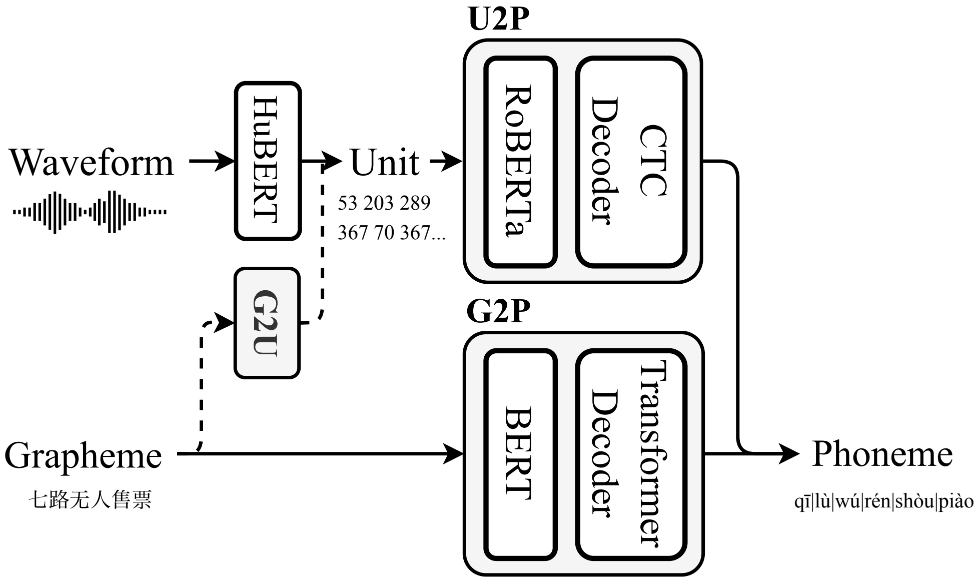


Figure 8.1: Architecture overview

8.4.1 Speech Units

We use the HuBERT model pretrained on 960 hours of English speech from LibriSpeech [73] to extract speech representation, regardless of the target language. Following the textless NLP [155], we use the representation from the 6th layer as it yields the best ABX score in phoneme categorization. Presumably a HuBERT model pretrained on large speech corpora from the target languages, namely Chinese and Japanese in this work, can yield better representation. The speech representations are K-Means clustered to 500 clusters, and the indices of the cluster centroids are used as speech units.

8.4.2 G2P

The G2P network is a transformer encoder-decoder architecture. The encoder is the pretrained BERT [68] model to extract contextual representations of the grapheme inputs. We use the `bert-base-chinese` for Chinese and `cl-tohoku/bert-base-japanese` for Japanese, both of which are available in HuggingFace Transformers [208]. The decoder is a transformer implemented in ESPNet [194] containing 6 self-attention encoder layers, each having 4 heads, an attention dimension of 256 and a 2048-dim position-wise feedforward layer. During training, the parameters of the BERT model are kept frozen. We take a weighted sum of the hidden representations from all layers and tune only the weights. The decoder is a randomly initialized transformer network and is trained from scratch.

8.4.3 U2P

The U2P network is a transformer-based encoder trained with a connectionist temporal classification (CTC) loss. The encoder is a RoBERTa network [69] implemented in fairseq [150] that is pretrained on the speech units.

8.4.4 G2U

The G2P and U2P network do not fully address the G2PU problem because they rely on a parallel waveform which is not usually available for arbitrary text. In such cases, we use a G2U model to predict acoustic units from graphemes, leveraging the pronunciation knowledge learned from the parallel speech-text training set. We use `transformer_iwslt_de_en`, a neural machine translation model implemented in Fairseq [150] and select the best model based on BiLingual Evaluation Understudy (BLEU) score. Although the validation BLEU score ranging from 30 to 40 seems low, preliminary experiments show that the audio synthesized using the generated unit sequences is well-intelligible.

8.5 Experiment Setup

We conduct experiments on Mandarin Chinese and Japanese. For Chinese, we use AISHELL-3 dataset [222], which contains 85 hours of speech from 218 native Chinese mandarin speakers and their transcripts in Chinese characters and Mandarin Chinese *pinyin*. The character transcripts are used as grapheme transcripts and the *pinyin* transcripts are used as the groundtruth phoneme transcripts. The tone mark is placed on vowels and the two combined are considered as one symbol. We use the original train split for training and randomly sample 20% utterances from the test split for validation and randomly sample 1k utterances from the remaining for testing.

For Japanese, we use Corpus of Spontaneous Japanese (CSJ), which contains about 650 hours of spontaneous speech by native speakers [100] along with their transcripts in Japanese text and in *katakana*. The Japanese text transcripts are used as grapheme transcripts and the *katakana* transcripts are used as the groundtruth phoneme transcripts. We preprocess the dataset using the CSJ recipe provided in ESPnet to split and segment the audio and filter out noisy utterances. The scripts create three splits: `train_nodup`, `train_dev`, and `eval1`. We randomly sample 100 hours of speech data from the `train_nodup` split for training and use `train_dev` and `eval1` splits for validation and testing respectively. The *katakana* transcripts contain long vowels denoted by "ー" that can be any one of the five Japanese vowels depending on its preceding *katakana*. We additionally substitute the long vowel symbol into the correct vowels [7]. For example "コー" would be expanded to "コウ".

8.5.1 Teacher G2P

For Chinese, we generate the teacher phoneme transcripts using Montreal Forced Aligner (MFA) [210] and G2PW [103]. MFA requires a lexicon containing character pronunciations in *pinyin* for which we use CC-CEDICT dictionary[8]. We train the G2PU model on the phoneme transcripts generated by MFA/G2PW and evaluate the model on the groundtruth phoneme transcripts. For Japanese, we obtain the teacher phoneme transcripts using MFA and Pykakasi[9], a python wrapper for Kakasi [223]. We use the dictionary provided in CSJ for MFA training. Similar to Chinese, the G2PU is trained on the generated phoneme transcripts.

8.5.2 Training and Evaluation

The G2PU model is trained for 150 epochs until convergence and the checkpoint with the best validation accuracy is used for evaluation. The evaluation metric is PER, which is the edit distance between the generated phoneme transcripts and the groundtruth transcripts.

[7] `en.wikibooks.org/wiki/Japanese/Kana#Hiragana`

[8] `https://cc-cedict.org/wiki`

[9] `https://github.com/miurahr/pykakasi`

Table 8.2: PER of phoneme transcripts generated by G2PU evaluated against groundtruth phonemes. ZH is the result for Chinese and JP is for Japanese.

		Teacher		G2PU		G2PU-PU	
Lang	Teacher	dev	test	dev	test	dev	test
ZH	MFA	6.8	5.9	**4.9**	**4.2**	5.1	4.3
	G2PW	2.7	2.3	**2.5**	**1.8**	**2.5**	**1.8**
JP	MFA	2.4	2	**1.9**	**1.5**	2.3	1.9
	Pykakasi	5.2	5.0	**4.5**	**4.6**	4.9	4.8

Table 8.3: PER of phoneme transcripts generated by G2PU and G2PU with audio waveform.

		G2PU		G2PU-WAV	
Lang	Teacher	dev	test	dev	test
ZH	MFA	4.9	4.2	4.9	4.2
JP	Pykaksi	4.5	4.6	4.5	4.6

8.6 Results

8.6.1 Phoneme Transcription Performance

We train the G2PU model using teacher phoneme transcripts generated by G2P tools. The results are shown in Table 8.2. By comparing Teacher (MFA, G2PW and Pykakasi) and G2PU, We observe that although the model is trained on Teacher transcripts, additional acoustic information helps the G2PU model to achieve a lower PER in both Chinese and Japanese compared to its teacher G2P. The Chinese experiments exhibit a larger gain than the Japanese experiments. Besides the language differences, the quality of the speech may also affect the model's performance, as the CSJ contains spontaneous speech, which has more irregular expression than the read speech in AISHELL-3.

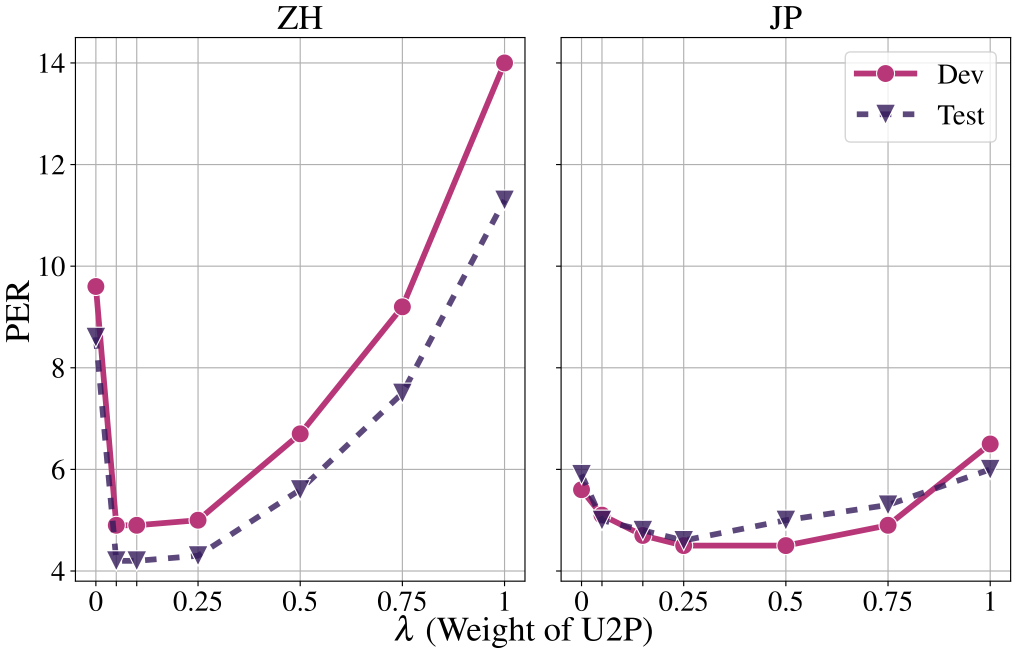


Figure 8.2: Performance over different weight of U2P model.

8.6.2 Weight of U2P

During the joint decoding of the U2P and G2P networks, the beam search takes a weighted sum of the score of the running hypotheses from both networks. We assign the weight λ to the U2P model and $1-\lambda$ to the G2P model and vary the weight value from 0 to 1 to plot the PER curves of G2PU in Chinese (teacher transcripts generated using G2PW) and Japanese (teacher transcripts generated using PYKAKASI), in Figure 8.2. On the leftmost side where $\lambda = 0$, the G2PU model uses only the G2P network, and on the rightmost side where $\lambda = 1$, the G2PU model uses only the U2P network. As shown, neither the G2P nor U2P network alone results in satisfactory transcription. Instead, the best performance is achieved at $\lambda = 0.05$ for Chinese and $\lambda = 0.25$ for Japanese. In the case of Chinese, the high error rate of G2P at $\lambda = 0$ is partly due to the limited amount of training data available. A closer examination of the G2P outputs shows that the G2P network terminates some predictions before finishing converting all the grapheme tokens. However, when additional acoustic information is integrated, even with $\lambda = 0.05$, the early termination issue disappears.

Table 8.4: Example transcripts generated by G2PU. “REF” denotes the groundtruth phoneme transcripts. “MFA” and “Pykakasi” denote the transcripts generated by existing G2P tools that are used to train the G2PU model. The incorrect phonemes are underlined.

ZH	
Grapheme	来自网购的竞争是原因之一
REF	lái zì wǎng gòu de jìng zhēng shì yuán yīn zhī yī
MFA	lái zì wǎng gòu d**í** jìng zhēng sh**i** yuán yīn zhī yī
G2P	lái zì wǎng gòu de jìng zhēng sh**i** yuán yīn zhī yī
U2P	lái zì wǎng gòu d**í** j**ià**n zhēng shì yuán y**ǐ**n zhī yī
G2PU	lái zì wǎng gòu de jìng zhēng shì yuán yīn zhī yī
Grapheme	以创造股东价值为主要目标
REF	yǐ chuàng zào gǔ dōng jià zhí wéi zhǔ yào mù biāo
MFA	yǐ chuàng zào gǔ dōng jià zhí w**è**i zhǔ yào mù biāo
G2P	yǐ chuàng zào g**u** dōng jià zhí wéi zhǔ yào mù biāo
U2P	yǐ chu**ā**ng zào gǔ dōng jià zhí w**è**i zhǔ yà**ng** mù biāo
G2PU	yǐ chuàng zào gǔ dōng jià zhí wéi zhǔ yào mù biāo
JP	
Grapheme	二通りの自動要約文を生成
REF	フタトオリノジドウヨウヤクブンオセイセイ
Pykakasi	ニトオリノジドウヨウヤクブンヲセイセイ
G2P	ニトオリノジ**ド**ウヨウヤクブンヲセイセイ
U2P	フタトオリノジドウイヤクモヲセイセイ
G2PU	フタトオリノジドウヨウヤクブンヲセイセイ
Grapheme	クーボビーの実験ではレベルの上昇時
REF	クーボビーノジッケンデワレベルノジョウショウジ
Pykakasi	クーボビーノジッケンデハレベルノジョウショウトキ
G2P	クーボビーノジッケンデハレベルノジョウショウトキ
U2P	クゴミノジッケンデハレベルノジョウショウジ
G2PU	クーボビーノジッケンデハレベルノジョウショウジ

8.6.3 G2P performance with Pseudo Units

We evaluate the G2PU model using G2U-generated pseudo acoustic units in place of the the real HuBERT units from waveforms. The results are shown in the “G2P-PU” column in Table 8.2. Using the generated pseudo units during the inference degrades the quality of the transcripts but G2PU-PU still outperforms the teacher G2P tool especially when the teacher G2P tools (ZH-MFA and JP-Pykakasi) are not very good.

8.6.4 G2P performance with Waveform

We additionally finetune a Wav2vec2-CTC ASR model that transcribes the waveform into phonemes. The Chinese Wav2vec2 is pretrained on 960 hours of Chinese speech sampled from United Nations Proceedings Speech [96] and the Japanese Wav2vec2 is pretrained on 960 hours of Japanese speech sampled from LaboroTVSpeech [97]. We conducted experiments on the ZH-MFA teacher and JP-Pykakasi G2PU's. These choices were driven by their higher error rates, thereby providing a greater scope for improvement if any. We repeat the G2PU experiments but replace the U2P network with the Wav2vec2-CTC model and show the PER result in the G2PU-WAV row in Table 8.3. The best performance is achieved at $\lambda = 0.05$ and $\lambda = 0.25$ for Chinese and Japanese, which is identical to using the units. Interestingly, the PER using the Wav2vec-CTC network is nearly the same as using U2P even though the waveform contains richer information than the units. This finding indicates acoustic units encode enough information to regularize the G2P predictions.

8.6.5 Qualitative Study on Transcripts

In Table 8.4, we show example phoneme transcripts of speech generated by the G2PU networks along with the groundtruth transcripts from Chinese and Japanese datasets. "REF" rows are the groundtruth phoneme transcripts. MFA and Pykakasi rows are the transcripts generated by teacher G2P tools and are used to train the G2PU networks. We underline the incorrectly predicted phonemes compared to the "REF" transcripts.

As we can see from "MFA" and "Pykakasi" rows, the teacher transcripts contain mistakes. In Chinese, MFA learns an acoustic model from parallel speech and text, then selects pronunciations from the lexicon. However, it mistakenly assigns "dí" to "的", "shi" to "是", and "wèi" to "为" because the pronunciation by the speaker is ambiguous. The G2P and the U2P networks trained on MFA transcripts partially inherit the mistakes from MFA transcripts and partially correct them. For example, in the left column, G2P corrects the phonemes for “的” using context information and U2P corrects the phonemes for “是” using acoustic information. By properly setting the weights between the G2P and U2P logits, the G2PU network is able to correct the mistakes in both networks and generate a transcript that is more accurate than the teacher transcripts.

Similar observations can be made in Japanese experiments. The *kanji* character “二” is sometimes pronounced “ニ” (/ni/) and sometimes “フタ” (/fu ta/), depending on context; similarly, “時” is sometimes pronounced “トキ” (/to ki/) and sometimes “ジ” (/ji/). The teacher Pykakasi knows many compound words containing these characters, but it does not know the compound words in these two test sentences. Therefore it must guess which pronunciation to use; it guesses incorrectly. Such mistakes are inherited by the G2P network trained on the Pykakasi transcripts. The U2P network on the other hand is able to correct these mistakes using the actual speech sound. However, the speaker pronounces the *kanji* “要” and the word “クーボビー” quickly, so the U2P mistranscribes these words as the short “イ” (/i/) and “クゴミ” (/ku go mi/) instead of the longer “ヨウ” (/yo u/) and “クーボビー” (/ku u bo bi i/). Again we see that a proper weight between the two network outputs can yield a better phoneme transcript than the teacher Pykakasi transcript.

The G2PU model is not perfect, however. In Japanese, the *hiragana* “は” is pronounced as “ハ” (/ha/) in most cases but is pronounced as “ワ” (/wa/) when used as a grammatical particle to mark the topic. Pykakasi does not distinguish between the two pronunciations and assigns “ハ” (/ha/) to all the occurrences of “は”. This mistake is learned by both the G2P and the U2P networks and we observe that the U2P network cannot correct the “ハ” (/ha/) to “ワ” (/wa/) using the speech units. This observation suggests the U2P model is not entirely acoustic; it learns a language model from the Pykakasi transcripts.

8.7 Summary

Grapheme-to-phoneme transducers (G2Ps) convert graphemes to phonemes. Forced alignment can be used to select from among alternate pronunciations, and the selected pronunciations can be used to re-train the G2P or the acoustic model, but we know of no existing system that simultaneously optimizes the G2P, the acoustic model, and the phone transcripts of the training data. In this paper, we propose G2PU, a joint CTC-attention model that is trained using teacher transcripts, and show that G2PU is able to output better phoneme transcripts than its teacher when conditioned on both the graphemes and the acoustic units.

CHAPTER 9

DISCUSSION

We have introduced six research projects that apply unsupervised speech technologies to improve speech processing systems on low-resource languages. This chapter discusses these projects' advantages and limitations.

9.1 On Resource-Limited Self-Supervised Learning

Since the success of SSL methods in natural language processing (NLP) [68, 69], computer vision (CV) [224] and Speech [10, 11, 12], various methods have been proposed to improve it. In speech processing domain, for example, XLSR [40] improves WAV2VEC2 [11] by incorporating additional data from different languages; VG-HUBERT [88] improves HUBERT by incorporating additional paired speech-image data; WAVLM [225] improves HUBERT by incorporating additional denoising objectives. These methods improve the quality of SSL representation, assuming additional data is available, whereas the reverse scenario, where data is limited, is less investigated. Differing from these existing methods, DIFFS4L proposed in Chapter 4 aims to improve SSL in a resource-limited setting. The amount of data involved is strictly controlled to be 100 hours of raw speech data without text labels. The seed SSL, the refined SSL models, the diffusion-based speech synthesizers, and the HIFIGAN vocoder are all trained on 100 hours of real data. Such data constraints underscore both the advantages and disadvantages of DIFFS4L.

The advantages of DIFFS4L are twofold. First, DIFFS4L shows that it extracts better speech representations compared to baseline SSL methods when trained on equivalent amounts of data. Experimental findings in Chapter 4 suggest that diffusion-based speech synthesizers can learn the most possible patterns in the speech and the resulting synthetic speech data with speaker and content artificially perturbed exhibit a closer resemblance to the real speech distribution than the original 100-hour data. This observation suggests the better information efficiency of DIFFS4L. Second, DIFFS4L uses synthetic data to augment the original 100 hours of real speech data, and thus is able to avoid the privacy and data security issues from the introduction of the additional real data.

The disadvantages of DIFFS4L originate from the data limitation. As observed from Table 4.1, using 960 hours of real speech is still better than using 100 hours of real speech plus 860 hours of synthetic speech data, because the real data drawn from the true speech distribution has a lower bias than the synthetic data. Besides, the training pipeline is overly lengthy, requiring the training of two diffusion models the training of two SSL models and the generation of around 1000 hours of synthetic speech, which is not efficient in practice.

9.2 On Two-Stage Modality Matching

Given the modality matching has been shown successful in unsupervised ASR [2], it is natural to question whether the unsupervised ASR can be applied to train an unsupervised TTS where parallel speech-text is unavailable. The project introduced in Chapter 5 is the first affirmative response to this question. In this project, a two-stage unsupervised TTS system is designed, which consists of an unsupervised ASR module to transcribe the speech into pseudo text labels and a traditional supervised TTS trained using the pseudo text labels and the speech. This system is trained and evaluated on a number of languages and the synthesized speech exhibits high intelligibility. However, this system still has limitations, with the major limitation being the cascading design containing the disjoint ASR and TTS systems. Firstly, while the pseudo text labels generated by the ASR subsystem determine the overall performance of the supervised system, there is no feedback mechanism to the ASR system to constrain error propagation. Secondly, the proposed cascaded system complicates the training process, and end-to-end designs of unsupervised TTS systems remain unexplored. Another limitation comes from the specific choice of TACOTRON2 as the supervised TTS model in this project. Experimental results show that the TACOTRON2 has convergence issues on certain languages and both supervised and unsupervised

TACOTRON2 produce synthetic speech with noticeable artifacts. Other TTS systems with stabler training processing can be explored.

9.3 On Unsupervised Transfer Learning

The difficulty of building a working low-resource speech processing system stems from data sparsity; we are using a limited amount of data samples to estimate the entire speech-text distribution. Apparently, prior knowledge about the target space is necessary to achieve performance gain under such limitations. Prior knowledge can come from a related domain. The project in Chapter 6 transfers phonetic prior knowledge learned from multiple languages to the ASR task on unseen languages. WAVPROMPT in Chapter 7 transfers semantic prior knowledge learned from large text corpora to answer SLU questions.

Prior knowledge can be classified as either theory-driven or data-driven, *i.e.*, summarized by experts or learned from data. The projects introduced in Chapter 6 and in Chapter 7 can be considered as incorporating these two types of prior knowledge, respectively.

In Chapter 6, two types of prior knowledge summarized by linguists, *i.e.*, the phylogenetic relationships between languages and the phoneme inventory, are introduced to a joint CTC-based Transformer model. The language embeddings extracted from the linguistic knowledge exhibit patterns that match human heuristic as shown in Figure 6.3; for example, although Spanish never appears in the training data, the model, relying on prior knowledge, assumes it to be a language similar to Portuguese, as indicated by the distance between the two in Figure 6.3. Consequently, it adapts itself using the Spanish language embedding to predict phonemes in a manner that aligns closely with Portuguese. The limitation of this work mainly stems from the language metadata involved. The metadata of phylogenetic information and the phoneme inventory are both quite abstract, lacking more specific knowledge about the training and the testing languages, such as syntax and grammar. With the advancement of prompt engineering technology and large language modeling, it might be interesting to see if LLMs can perform zero-shot phonetic recognition with more detailed text descriptions describing the syntactic and grammatical rules of the testing language.

In Chapter 7, WavPrompt incorporates data-driven prior knowledge into an SLU system through a frozen pretrained language model. WavPrompt system consists of an audio encoder and a frozen language model. The audio encoder is pretrained as part of an ASR system so that it learns to convert the speech in the task demonstrations into embeddings digestible to the language model. After pretraining, the entire framework is frozen and ready to perform few-shot learning upon seeing the demonstrations. Notably, the training and the testing tasks are different; the training task, ASR, requires the system to transcribe the speech literally, whereas the test task, SLU, requires the system to answer questions regarding the speech content. In this sense, WavPrompt can be considered as an unsupervised SLU system as it does not require labeled SLU data. The different question prompts serve as adapters to adapt the language model to perform different tasks. WavPrompt is among the earliest attempts to build a prompted-based SLU model. The largest limitation of WavPrompt is it only performs intention classification and is unable to perform generation-based question answering. This can be due to the limited training data and the limited size of the language model. The training data is only 960 hours of speech data in the LibriSpeech corpus, and the language model involved is GPT2 with only 117 Million parameters. Using language models with more parameters and stronger text-generation capacity is a future direction to explore.

9.4 On Unsupervised Multimodal Learning

Prior knowledge to improve the low-resource models can also come from a different modality. In Chapter 8, we show that by combining the text and speech modalities, G2PU can generate improved phoneme transcripts compared to its teacher. As suggested in the qualitative results in Sec 8.6.5, G2PU, leveraging the additional speech modality, corrects errors predicted by the text-based G2P transducers. The entire training process does not involve groundtruth phoneme transcripts as targets. In this sense, G2PU can be classified as an unsupervised method for improving existing G2P tools. The main limitation of G2PU is that it still requires groundtruth phoneme transcripts to select the best model during evaluation, and there does not exist an unsupervised metric to evaluate the generated phoneme transcripts.

The lack of groundtruth transcripts for evaluation limits languages available for experiments and also limits its application in practice. There are two solutions to this limitation. One is to collect a test set with groundtruth phoneme transcript for each target language. The other is to design an unsupervised metric for phoneme transcripts, which can be a future exploration direction.

CHAPTER 10

CONCLUSION

This thesis explores applying unsupervised speech technologies to improve speech processing systems. In Chapter 3, we study the transferability of SSL models trained using monolingual, cross-lingual and multilingual data and observe that even the monolingually trained WAV2VEC2 models have achieved decent performance in unseen languages. In Chapter 4, we propose DIFFS4L to improve the data efficiency of the SSL models such as WAV2VEC2 and HUBERT, using synthetic data generated by diffusion models. In Chapter 5, we introduce the first attempt to build an unsupervised TTS system using unsupervised ASR methods. In Chapter 6, we use external linguistic knowledge to improve the zero-shot cross-lingual phonetic recognition systems without using any data from the testing languages. In Chapter 7, we propose WAVPROMPT to leverage the few-shot learning ability of pretrained language models to perform SLU without any labeled SLU data. In Chapter 8, we propose G2PU that improves the prediction of a G2P transducer by incorporating additional acoustic information without access to groundtruth phonetic transcripts during training.

These projects demonstrate that although unsupervised learning does not provide the same benefit as an increased amount of labeled data, nevertheless, unsupervised learning can significantly improve the performance of speech recognition, speech synthesis, and speech understanding in under-resourced application scenarios. All the projects in this thesis encode prior knowledge into neural representations for knowledge transfer from one task to another. These neural representations can be classified into three broad categories: (1) trained model parameters, as in Chapter 3, Chapter 5, Chapter 7 and Chapter 8 (2) synthetic data, as in Chapter 4, or (3) explicit encoding of linguistic scientific knowledge about the target domain as in Chapter 6. All three types of knowledge transfer have been demonstrated to reduce error rates of speech processing systems.

There have been many studies on unsupervised speech technologies. The methods included in this thesis are just small attempts to improve speech processing systems on low-resource languages. I hope that this thesis could invite other researchers to explore unsupervised speech technologies and bring the benefits of modern speech processing systems to a wider user base.

REFERENCES

[1] D. M. Eberhard, G. F. Simons, and C. D. Fennig, "Ethnologue: Languages of the world. twenty-sixth edition. dallas, texas: Sil international," 2023. [Online]. Available: http://www.ethnologue.com

[2] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 826–27 839, 2021.

[3] J. Ni, L. Wang, H. Gao, K. Qian, Y. Zhang, S. Chang, and M. A. Hasegawa-Johnson, "Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition," in *Interspeech*, 2022.

[4] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *Proceedings of the 23rd international conference on Machine learning*, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:9901844

[5] A. Graves, "Sequence transduction with recurrent neural networks," *ArXiv*, vol. abs/1211.3711, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:17194112

[6] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning.* PMLR, 2016, pp. 173–182.

[7] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[8] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and LSTM encoder decoder models for ASR," in *ICASSP*, 2019, pp. 8–15.

[9] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *ArXiv*, vol. abs/1904.08779, 2019.

[10] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019, pp. 3465–3469.

[11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[12] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[13] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[14] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[15] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu1, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2018.

[16] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2020.

[18] Y. A. Li, C. Han, and N. Mesgarani, "Styletts: A style-based generative model for natural and diverse text-to-speech synthesis," *ArXiv*, vol. abs/2205.15439, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249209998

[19] Y. A. Li, C. Han, V. S. Raghavan, G. Mischler, and N. Mesgarani, "Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models," *ArXiv*, vol. abs/2306.07691, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259145293

[20] C. Liu, S. Zhu, Z. Zhao, R. Cao, L. Chen, and K. Yu, "Jointly encoding word confusion network and dialogue context with bert for spoken language understanding," *ArXiv*, vol. abs/2005.11640, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218869853

[21] S. Arora, S. Dalmia, P. Denisov, X. Chang, Y. Ueda, Y. Peng, Y. Zhang, S. S. Kumar, K. Ganesan, B. Yan, N. T. Vu, A. W. Black, and S. Watanabe, "Espnet-slu: Advancing spoken language understanding through espnet," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7167–7171, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244714990

[22] S. Arora, S. Dalmia, X. Chang, B. Yan, A. W. Black, and S. Watanabe, "Two-pass low latency end-to-end spoken language understanding," in *Interspeech*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250526182

[23] V. Sunder, E. Fosler-Lussier, S. Thomas, H.-K. J. Kuo, and B. Kingsbury, "Tokenwise contrastive pretraining for finer speech-to-bert alignment in end-to-end speech-to-intent systems," in *Interspeech*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248085405

[24] Y. Ma, T. H. Nguyen, J. Ni, W. Wang, Q. Chen, C. Zhang, and B. Ma, "Auxiliary pooling layer for spoken language understanding," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258530329

[25] N. A. Tu, D. X. Hieu, T. M. Phuong, and N. X. Bach, "A bidirectional joint model for spoken language understanding," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258530062

[26] J. Wang, M. H. Radfar, K. Wei, and C. Chung, “End-to-end spoken language understanding using joint ctc loss and self-supervised, pretrained acoustic encoders,” *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258479837

[27] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” *arXiv preprint arXiv:1912.06670*, 2019.

[28] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[29] L. Wang, M. A. Hasegawa-Johnson, and C. D. Yoo, “A theory of unsupervised speech recognition,” *ArXiv*, vol. abs/2306.07926, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259144969

[30] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. J’egou, “Word translation without parallel data,” *ArXiv*, vol. abs/1710.04087, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3470398

[31] G. Lample, L. Denoyer, and M. Ranzato, “Unsupervised machine translation using monolingual corpora only,” *ArXiv*, vol. abs/1711.00043, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3518190

[32] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, “Unsupervised neural machine translation,” *ArXiv*, vol. abs/1710.11041, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3515219

[33] D.-R. Liu, K.-Y. Chen, H. yi Lee, and L.-S. Lee, “Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings,” *ArXiv*, vol. abs/1804.00316, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4553465

[34] C.-K. Yeh, J. Chen, C. Yu, and D. Yu, “Unsupervised speech recognition via segmental empirical output distribution matching,” *ArXiv*, vol. abs/1812.09323, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:56895485

[35] Y.-C. Chen, C.-H. Shen, S.-F. Huang, and H. yi Lee, "Towards unsupervised automatic speech recognition trained by unaligned speech and text only," *ArXiv*, vol. abs/1803.10952, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4458265

[36] K.-Y. Chen, C.-P. Tsai, D.-R. Liu, H. yi Lee, and L.-S. Lee, "Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden markov models," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201645152

[37] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[38] M. Gales, S. Young et al., "The application of hidden markov models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[39] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schluter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257365554

[40] A. Conneau, A. Baevski, R. Collobert, A. rahman Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Interspeech*, 2020.

[41] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[42] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[43] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[44] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[45] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising Diffusion Restoration Models." [Online]. Available: http://arxiv.org/abs/2201.11793

[46] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations." [Online]. Available: http://arxiv.org/abs/2011.13456

[47] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *International Conference on Machine Learning.* PMLR, 2021, pp. 8599–8608.

[48] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, and Y. Ren, "Prodiff: Progressive fast diffusion model for high-quality text-to-speech," *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[49] L. Sari, S. Thomas, and M. A. Hasegawa-Johnson, "Training spoken language understanding systems with non-parallel speech and text," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8109–8113, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:216271964

[50] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:204838007

[51] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[52] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[53] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu et al., "On decoder-only architecture for speech-to-text and large language model integration," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).* IEEE, 2023, pp. 1–8.

[54] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:249605738

[55] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, "Listen, think, and understand," *ArXiv*, vol. abs/2305.10790, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258762560

[56] F. Chen, M. Han, H. Zhao, Q. Zhang, J. Shi, S. Xu, and B. Xu, "X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages," *ArXiv*, vol. abs/2305.04160, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258558106

[57] E. Nachmani, A. Levkovitch, J. Salazar, C. Asawaroengchai, S. Mariooryad, R. Skerry-Ryan, and M. T. Ramanovich, "Lms with a voice: Spoken language modeling beyond speech tokens," *arXiv preprint arXiv:2305.15255*, 2023.

[58] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. T. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. Clark, L. E. Shafey, Y. Huang, K. S. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. A. Botha, J. Bradbury, S. Brahma, K. M. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crépy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. C. D'iaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. García, S. Gehrmann, L. González, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. R. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. H. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M.-L. Li, W. Li, Y. Li, J. Y. Li, H. Lim, H. Lin, Z.-Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, O. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. Ros, A. Roy, B. Saeta, R. Samuel, R. M. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. W. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, and Y. Wu, "Palm 2 technical report," *ArXiv*, vol. abs/2305.10403, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258740735

[59] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4225–4229, 2015.

[60] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Grapheme-to-phoneme conversion with convolutional neural networks," *Applied Sciences*, vol. 9, no. 6, p. 1143, 2019.

[61] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Transformer based grapheme-to-phoneme conversion," *arXiv preprint arXiv:2004.06338*, 2020.

[62] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "Byt5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[63] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[64] M. Řezáčková, J. Švec, and D. Tihelka, "T5g2p: Using text-to-text transfer transformer for grapheme-to-phoneme conversion," 2021.

[65] Q. Wang, "A method of polyphone disambiguation based on semantic extension," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4. IEEE, 2021, pp. 1752–1756.

[66] M. Rezaei, N. Nayeri, S. Farzi, and H. Sameti, "Multi-module g2p converter for persian focusing on relations between words," *arXiv preprint arXiv:2208.01371*, 2022.

[67] A. Ploujnikov and M. Ravanelli, "Soundchoice: Grapheme-to-phoneme models with semantic disambiguation," in *Interspeech*, 2022.

[68] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019. [Online]. Available: https://doi.org/10.18653/v1/n19-1423 pp. 4171–4186.

[69] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[70] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, "ContentVec: An improved self-supervised speech representation by disentangling speakers," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022. [Online]. Available: https://proceedings.mlr.press/v162/qian22b.html pp. 18 003–18 017.

[71] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[72] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[73] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[74] L. Wang, J. Ni, H. Gao, J. Li, K. C. Chang, X. Fan, J. Wu, M. Hasegawa-Johnson, and C. Yoo, "Listen, decipher and sign: Toward unsupervised speech-to-sign language recognition," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, July 2023. [Online]. Available: https://aclanthology.org/2023.findings-acl.424 pp. 6785–6800.

[75] P. Żelasko, L. Moro-Velázquez, M. Hasegawa-Johnson, O. Scharenborg, and N. Dehak, "That sounds familiar: an analysis of phonetic representations transfer across languages," in *Interspeech*, 2020, pp. 3705–3709.

[76] X. Li, S. Dalmia, J. Li, M. Lee, P. Littell, J. Yao, A. Anastasopoulos, D. R. Mortensen, G. Neubig, A. W. Black et al., "Universal phone recognition with a multilingual allophone system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8249–8253.

[77] H. Gao, J. Ni, Y. Zhang, K. Qian, S. Chang, and M. Hasegawa-Johnson, "Zero-shot cross-lingual phonetic recognition with external language embedding." in *Interspeech*, 2021, pp. 1304–1308.

[78] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4909–4913.

[79] X. Li, S. Dalmia, D. Mortensen, J. Li, A. Black, and F. Metze, "Towards zero-shot learning for automatic phonemic transcription," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8261–8268.

[80] G. I. Winata, G. Wang, C. Xiong, and S. Hoi, "Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition," *arXiv preprint arXiv:2012.01687*, 2020.

[81] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language and speech recognition," in *IEEE Proceedings on Automatic Speech Recognition and Understanding*, 2017, pp. 265–271.

[82] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2018, pp. 521–527.

[83] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively multilingual adversarial speech recognition," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 96–108.

[84] B. Li, T. Sainath, K. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4749–4753, 2018.

[85] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4904–4908.

[86] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," in *Proc. Interspeech*, 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2858 pp. 2130–2134.

[87] X. Li, S. Dalmia, A. Black, and F. Metze, "Multilingual speech recognition with corpus relatedness sampling," in *INTERSPEECH*, 2019.

[88] P. Peng and D. F. Harwath, "Word discovery in visually grounded, self-supervised speech models," in *Interspeech*, 2022.

[89] Y.-J. Shih, H.-F. Wang, H.-J. Chang, L. Berry, H. yi Lee, and D. F. Harwath, "Speechclip: Integrating speech with pretrained vision and language model," *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 715–722, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252683634

[90] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *ArXiv*, vol. abs/2206.06488, 2022.

[91] J. Ao, R. Wang, L. Zhou, S. Liu, S. Ren, Y. Wu, T. Ko, Q. Li, Y. Zhang, Z. Wei, Y. Qian, J. Li, and F. Wei, "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing," *ArXiv*, vol. abs/2110.07205, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238856828

[92] S. Communication, L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. ElSahar, H. Gong, K. Heffernan, J. Hoffman, C. Klaiber, P. Li, D. Licht, J. Maillard, A. Rakotoarison, K. R. Sadagopan, G. Wenzek, E. Ye, B. Akula, P.-J. Chen, N. E. Hachem, B. Ellis, G. M. Gonzalez, J. Haaheim, P. Hansanti, R. Howes, B. Huang, M.-J. Hwang, H. Inaguma, S. Jain, E. Kalbassi, A. Kallet, I. Kulikov, J. Lam, S.-W. Li, X. Ma, R. Mavlyutov, B. Peloquin, M. Ramadan, A. Ramakrishnan, A. Sun, K. M. Tran, T. Tran, I. Tufanov, V. Vogeti, C. Wood, Y. Yang, B. Yu, P. Y. Andrews, C. Balioglu, M. R. Costa-jussà, O. Çelebi, M. Elbayad, C. Gao, F. Guzm'an, J. T. Kao, A. Lee, A. Mourachko, J. M. Pino, S. Popuri, C. Ropers, S. Saleem, H. Schwenk, P. Tomasello, C. Wang, J. Wang, and S. Wang, "Seamlessm4t: Massively multilingual&multimodal machine translation," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261064881

[93] T. Scialom, P. Bordes, P.-A. Dray, J. Staiano, and P. Gallinari, "What bert sees: Cross-modal transfer for visual question generation," in *International Conference on Natural Language Generation*, 2020.

[94] M. Tsimpoukelli, J. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," in *Neural Information Processing Systems*, 2021.

[95] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[96] K. Chay, C. Elizalde, and M. Ziemski, "United Nations Proceedings Speech," 2023. [Online]. Available: https://hdl.handle.net/11272.1/AB2/3LTQ01

[97] S. Ando and H. Fujihara, "Construction of a large-scale japanese asr corpus on tv recordings," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6948–6952.

[98] T. Schultz, N. T. Vu, and T. Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8126–8130.

[99] K. Walker et al., "GALE Phase 4 Arabic Broadcast News Speech LDC2018S05," *Web Download. Philadelphia: Linguistic Data Consortium*, 2018.

[100] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[101] "Russian librispeech (ruls)," https://www.openslr.org/96/, accessed: 2023-04-03.

[102] M. Hasegawa-Johnson, L. Rolston, C. Goudeseune, G.-A. Levow, and K. Kirchhoff, "Grapheme-to-phoneme transduction for cross-language asr," in *International Conference on Statistical Language and Speech Processing*. Springer, 2020, pp. 3–19.

[103] Y.-C. Chen, Y.-C. Steven, Y.-C. Chang, and Y.-R. Yeh, "g2pW: A Conditional Weighted Softmax BERT for Polyphone Disambiguation in Mandarin," in *Proc. Interspeech 2022*, 2022, pp. 1926–1930.

[104] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[105] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[106] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *arXiv preprint arXiv:2205.14217*, 2022.

[107] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf pp. 8780–8794.

[108] M. Baradad Jurjo, J. Wulff, T. Wang, P. Isola, and A. Torralba, "Learning to see by looking at noise," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/14f2ebeab937ca128186e7ba876faef9-Paper.pdf pp. 2556–2569.

[109] A. Jahanian, X. Puig, Y. Tian, and P. Isola, "Generative models as a data source for multiview representation learning," in *International Conference on Learning Representations*, 2021.

[110] Y. Wu, Z. Wang, D. Zeng, Y. Shi, and J. Hu, "Synthetic data can also teach: Synthesizing effective data for unsupervised visual representation learning," 2022. [Online]. Available: https://arxiv.org/abs/2202.06464

[111] H. Kataoka, R. Hayamizu, R. Yamada, K. Nakashima, S. Takashima, X. Zhang, E. J. Martinez-Noriega, N. Inoue, and R. Yokota, "Replacing labeled real-image datasets with auto-generated contours," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 232–21 241.

[112] C. Gan, J. Schwartz, S. Alter, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano et al., "Threedworld: A platform for interactive multi-modal physical simulation," in *Annual Conference on Neural Information Processing Systems*, 2021.

[113] H. Mikami, K. Fukumizu, S. Murai, S. Suzuki, Y. Kikuchi, T. Suzuki, S.-i. Maeda, and K. Hayashi, "A scaling law for synthetic-to-real transfer: How much is your pre-training effective?" *arXiv preprint arXiv:2108.11018*, 2021.

[114] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1278–1286.

[115] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7249–7255.

[116] P. Chattopadhyay, K. Sarangmath, V. Vijaykumar, and J. Hoffman, "Pasta: Proportional amplitude spectrum training augmentation for syn-to-real domain generalization," *arXiv preprint arXiv:2212.00979*, 2022.

[117] M.-C. Tsai and S.-D. Wang, "Self-supervised image anomaly detection and localization with synthetic anomalies," *Available at SSRN 4264542*, 2022.

[118] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[119] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 635–12 644.

[120] C. R. De Souza, A. Gaidon, Y. Cabon, and A. M. Lopez, "Procedural generation of videos to train deep action recognition networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2017, pp. 2594–2604.

[121] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," *Int. J. Comput. Vision*, vol. 129, no. 7, p. 2264–2287, jul 2021. [Online]. Available: https://doi.org/10.1007/s11263-021-01467-7

[122] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901–2910.

[123] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[124] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik et al., "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9339–9347.

[125] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9068–9079.

[126] C. Downey, L. Liu, X. Zhou, and S. Steinert-Threlkeld, "Learning to translate by learning to communicate," *ArXiv*, vol. abs/2207.07025, 2022.

[127] S. Yao, M. Yu, Y. Zhang, K. R. Narasimhan, J. B. Tenenbaum, and C. Gan, "Linking emergent and natural languages via corpus transfer," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=49A1Y6tRhaq

[128] S. Steinert-Threlkeld, X. Zhou, Z. Liu, and C. Downey, "Emergent communication fine-tuning (ec-ft) for pretrained language models," in *Emergent Communication Workshop at ICLR 2022*, 2022.

[129] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. F. Diamos, E. Elsen, J. Engel, L. J. Fan, C. Fougner, A. Y. Hannun, B. Jun, T. X. Han, P. LeGresley, X. Li, L. Lin, S. Narang, A. Ng, S. Ozair, R. J. Prenger, S. Qian, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, A. Sriram, C.-J. Wang, Y. Wang, Z. Wang, B. Xiao, Y. Xie, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2 : End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2015.

[130] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazar'e, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 215–222, 2020.

[131] J. Zhao, G. Haffar, and E. Shareghi, "Generating synthetic speech from spokenvocab for speech translation," *arXiv preprint arXiv:2210.08174*, 2022.

[132] K. Li, S. Li, X. Lu, M. Akagi, M. Liu, L. Zhang, C. Zeng, L. Wang, J. Dang, and M. Unoki, “Data augmentation using mcadams-coefficient-based speaker anonymization for fake audio detection,” in *Proc. INTERSPEECH*, 2022.

[133] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, “Back-translation-style data augmentation for end-to-end asr,” in *2018 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2018, pp. 426–433.

[134] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, “Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition,” in *2018 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2018, pp. 477–484.

[135] J. Li, R. T. Gadde, B. Ginsburg, and V. Lavrukhin, “Training neural speech recognition systems with synthetic speech augmentation,” *ArXiv*, vol. abs/1811.00707, 2018.

[136] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, “Generating synthetic audio data for attention-based speech recognition systems,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 7069–7073.

[137] L. P. Violeta, D. Ma, W.-C. Huang, and T. Toda, “Intermediate fine-tuning using imperfect synthetic speech for improving electrolaryngeal speech recognition,” *arXiv preprint arXiv:2211.01079*, 2022.

[138] Z. Jin, X. Xie, M. Geng, T. Wang, S. Hu, J. Deng, G. Li, and X. Liu, “Adversarial data augmentation using vae-gan for disordered speech recognition,” *arXiv preprint arXiv:2211.01646*, 2022.

[139] P. K. Krug, P. Birkholz, B. Gerazov, D. R. van Niekerk, A. Xu, and Y. Xu, “Articulatory Synthesis for Data Augmentation in Phoneme Recognition,” in *Proc. Interspeech 2022*, 2022, pp. 1228–1232.

[140] R. Zevallos, N. Bel, G. Cámbara, M. Farrús, and J. Luque, “Data augmentation for low-resource quechua asr improvement,” *arXiv preprint arXiv:2207.06872*, 2022.

[141] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, “Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 5674–5678.

[142] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning.* PMLR, 2018, pp. 5180–5189.

[143] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 461–11 471.

[144] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=NsMLjcFaO8O

[145] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, N. Dehak, and W. Chan, "Wavegrad 2: Iterative refinement for text-to-speech synthesis," in *Interspeech*, 2021.

[146] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *ArXiv*, vol. abs/2009.09761, 2020.

[147] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, "BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=L7wzpQttNO

[148] R. Huang, M. W. Y. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao, "Fastdiff: A fast conditional diffusion model for high-quality speech synthesis," in *International Joint Conference on Artificial Intelligence*, 2022.

[149] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," in *International Conference on Learning Representations*, 2019.

[150] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[151] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, https://github.com/facebookresearch/libri-light. pp. 7669–7673.

[152] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *ArXiv*, vol. abs/2012.03411, 2020.

[153] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. M. Pino, A. Baevski, A. Conneau, and M. Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *ArXiv*, vol. abs/2111.09296, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244270531

[154] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *Interspeech*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222310784

[155] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed et al., "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[156] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. NeurIPS*, 2022.

[157] Y. Wang, D. S. RJ Skerry-Ryan, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio et al., "Tacotron: Towards end-to-end speech synthesis," in *arXiv*, 2017. [Online]. Available: preprintarXiv:1703.10135

[158] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," 2018.

[159] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019.

[160] N.Li, S.Liu, Y.Liu, S.Zhao, and M.Liu, "Neural speech synthesis with transformer network," in *AAAI*, vol. 33, 2019, p. 6706–6713.

[161] J. Xu, X. Tan, Y. Ren, T. Qin, J. Li, S. Zhao, and T. Liu, "LRSpeech: Extremely low-resource speech synthesis and recognition," in *KDD*, 2020, pp. 2802–2812.

[162] K. Park and T. Mulc, "CSS10: A collection of single speaker speech datasets for 10 languages," 2019.

[163] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," 2021.

[164] P. K. Muthukumar and A. W. Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," 2014, pp. 2594–2598.

[165] A. H. Liu, T. Tu, H. Lee, and L. Lee, "Towards unsupervised speech recognition and synthesis with quantized speech representation learning," 2020, pp. 7259–7263.

[166] H. Zhang and Y. Lin, "Unsupervised learning for sequence-to-sequence text-to-speech for low-resource languages," 2020, pp. 3161–3165.

[167] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On generative spoken language modeling from raw audio," in *arXiv*, 2021. [Online]. Available: https://arxiv.org/pdf/2102.01192.pdf

[168] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Proc. Interspeech 2021*, 2021, pp. 3615–3619.

[169] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T.-A. Nguyen, M. Rivière, A. Mohamed, E. Dupoux, and W.-N. Hsu, "Text-free prosody-aware generative spoken language modeling," in *arXiv*, 2021. [Online]. Available: https://arxiv.org/pdf/2109.03264.pdf

[170] M. Hasegawa-Johnson, A. Black, L. Ondel, O. Scharenborg, and F. Ciannella, "Image2speech: Automatically generating audio descriptions of images," in *ICNLSSP*, 2017, p. 1–5.

[171] X. Wang, S. Feng, J. Zhu, M. Hasegawa-Johnson, and O. Scharenborg, "Show and speak: directly synthesize spoken description of images," in *icassp*, 2021.

[172] W.-N. Hsu, D. Harwath, T. Miller, C. Song, and J. Glass, "Text-free image-to-speech synthesis using learned segmental units," in *ACL-IJCNLP*, 2021, pp. 5284–5300.

[173] J. Effendi, S. Sakti, and S. Nakamura, "End-to-end image-to-speech generation for untranscribed unknown languages," *IEEE Access*, vol. 9, pp. 55 144–55 154, 2021.

[174] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Almost unsupervised text to speech and automatic speech recognition," 2019, pp. 5410–5419.

[175] C.-K. Yeh, J. Chen, C. Yu, and D. Yu, "Unsupervised speech recognition via segmental empirical output distribution matching," 2019.

[176] K.-Y. Chen, C.-P. Tsai, D.-R. Liu, H.-Y. Lee, and L. shan Lee, "Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden Markov models," 2019.

[177] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[178] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," 2018, pp. 4784–4788.

[179] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[180] K. Park and J. Kim, "g2pe," https://github.com/Kyubyong/g2p, 2019.

[181] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[182] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," 2020, pp. 7654–7658.

[183] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," 2020.

[184] K. Park and T. Mulc, "Css10: A collection of single speaker speech datasets for 10 languages," in *Interspeech*, 2019.

[185] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.

[186] H. Hammarström, R. Forkel, M. Haspelmath, and S. Bank, *Glottolog 4.3*, Jena, 2020. [Online]. Available: https://glottolog.org/accessed2021-03-30

[187] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[188] S. Moran, D. McCloy, and R. Wright, “Phoible online,” 2014.

[189] J. Li and M. Hasegawa-Johnson, “Autosegmental neural nets: Should phones and tones be synchronous or asynchronous?” in *Interspeech*, 2020.

[190] M. Hasegawa-Johnson, P. Jyothi, D. McCloy, M. Mirbagheri, G. di Liberto, A. Das, B. Ekin, C. Liu, V. Manohar, H. Tang, E. C. Lalor, N. Chen, P. Hager, T. Kekona, R. Sloan, , and A. K. Lee, “Asr for under-resourced languages from probabilistic transcription,” *IEEE/ACM Trans. Audio, Speech and Language*, vol. 25, no. 1, pp. 46–59, 2017.

[191] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Applying wav2vec2. 0 to speech recognition in various low-resource languages,” *arXiv preprint arXiv:2012.12121*, 2020.

[192] X. Li, S. Dalmia, D. R. Mortensen, F. Metze, and A. W. Black, “Zero-shot learning for speech recognition with universal phonetic model,” 2018.

[193] N. Oostdijk, “The spoken dutch corpus. overview and first evaluation.” in *LREC*. Athens, Greece, 2000, pp. 887–894.

[194] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen et al., “Espnet: End-to-end speech processing toolkit,” *Proc. Interspeech*, pp. 2207–2211, 2018.

[195] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang et al., “A comparative study on transformer vs RNN in speech applications,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[196] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE.” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[197] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey,” *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[198] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, “Calibrate before use: Improving few-shot performance of language models,” in *International Conference on Machine Learning.* PMLR, 2021, pp. 12 697–12 706.

[199] H. Gao, J. Ni, K. Qian, Y. Zhang, S. Chang, and M. A. Hasegawa-Johnson, “Wavprompt: Towards few-shot spoken language understanding with frozen language models,” in *Interspeech*, 2022.

[200] M. Tsimpoukelli, J. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, “Multimodal few-shot learning with frozen language models,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[201] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. abs/2101.00190, 2021.

[202] D. F. Harwath and J. R. Glass, “Deep multimodal semantic embeddings for speech and images,” *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 237–244, 2015.

[203] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, “Speech Model Pre-Training for End-to-End Spoken Language Understanding,” in *Proc. Interspeech 2019*, 2019, pp. 814–818.

[204] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, “SLURP: A spoken language understanding resource package,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://aclanthology.org/2020.emnlp-main.588 pp. 7252–7262.

[205] W.-N. Hsu, D. F. Harwath, C. Song, and J. R. Glass, “Text-free image-to-speech synthesis using learned segmental units,” in *ACL*, 2021.

[206] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” in *Proceedings of the 23rd Annual ACM Conference on Multimedia.* ACM Press, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390 pp. 1015–1018.

[207] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[208] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Online: Association for Computational Linguistics, Oct. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6 pp. 38–45.

[209] P. Żelasko, L. Moro-Vel'azquez, M. A. Hasegawa-Johnson, O. Scharenborg, and N. Dehak, "That sounds familiar: an analysis of phonetic representations transfer across languages," in *Interspeech*, 2020.

[210] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech*, 2017.

[211] M.-Y. Tsai, F. chiang Chou, and L. shan Lee, "Improved pronunciation modeling by properly integrating better approaches for baseform generation, ranking and pruning," in *ISCA Workshop on Pronunciation Modeling and Lexical Access (PMLA)*, Estes Park, Colorado, 2002, pp. 77–82.

[212] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Transformer based grapheme-to-phoneme conversion," in *Interspeech*, 2019.

[213] K. Park and S. Lee, "A neural grapheme-to-phoneme conversion package for mandarin chinese based on a new open benchmark dataset," *Proc. Interspeech 2020*, 2020. [Online]. Available: https://arxiv.org/abs/2004.03136

[214] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1215

[215] K. Park and S. Lee, "g2pM: A Neural Grapheme-to-Phoneme Conversion Package for Mandarin Chinese Based on a New Open Benchmark Dataset," in *Proc. Interspeech 2020*, 2020, pp. 1723–1727.

[216] H. Zhang, J. Yu, W. Zhan, and S. Yu, "Disambiguation of chinese polyphonic characters," in *The First International Workshop on MultiMedia Annotation (MMA2001)*, vol. 1. Citeseer, 2001, pp. 30–1.

[217] M. Řezáčková, J. Švec, and D. Tihelka, "T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion," in *Proc. Interspeech 2021*, 2021, pp. 6–10.

[218] Y. Shi, C. Wang, Y. Chen, and B. Wang, "Polyphone Disambiguation in Mandarin Chinese with Semi-Supervised Learning," in *Proc. Interspeech 2021*, 2021, pp. 4109–4113.

[219] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[220] T. Hori, S. Watanabe, and J. R. Hershey, "Joint ctc/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 518–529.

[221] H. Seki, T. Hori, S. Watanabe, N. Moritz, and J. Le Roux, "Vectorized beam search for ctc-attention-based speech recognition." in *INTERSPEECH*, 2019, pp. 3825–3829.

[222] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "AISHELL-3: A Multi-Speaker Mandarin TTS Corpus," in *Proc. Interspeech 2021*, 2021, pp. 2756–2760.

[223] "KAKASI - kanji kana simple inverter," 1999. [Online]. Available: http://kakasi.namazu.org/index.html.en

[224] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225039882

[225] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.