INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Improving grapheme-to-phoneme conversion by learning pronunciations from speech recordings

Manuel Sam Ribeiro, Giulia Comini, Jaime Lorenzo-Trueba

Amazon Alexa, TTS Research

{manuerib, gcomini, truebaj}@amazon.com

Abstract

The Grapheme-to-Phoneme (G2P) task aims to convert orthographic input into a discrete phonetic representation. G2P conversion is beneficial to various speech processing applications, such as text-to-speech and speech recognition. However, these tend to rely on manually-annotated pronunciation dictionaries, which are often time-consuming and costly to acquire. In this paper, we propose a method to improve the G2P conversion task by learning pronunciation examples from audio recordings. Our approach bootstraps a G2P with a small set of annotated examples. The G2P model is used to train a multilingual phone recognition system, which then decodes speech recordings with a phonetic representation. Given hypothesized phoneme labels, we learn pronunciation dictionaries for out-of-vocabulary words, and we use those to re-train the G2P system. Results indicate that our approach consistently improves the phone error rate of G2P systems across languages and amount of available data.

Index Terms: grapheme-to-phoneme, phone recognition, pronunciation modeling, low resource, text-to-speech

1. Introduction

The Grapheme-to-Phoneme (G2P) task aims to convert orthographic input, a sequence of *graphemes*, into a discrete phonetic representation, a sequence of *phonemes*. The ability to automatically convert graphemes into phonemes benefits speech processing systems such as Text-to-Speech (TTS) or Automatic Speech Recognition (ASR) by hypothesizing pronunciations for out-of-vocabulary words and generalizing beyond a finite, potentially small, manually-annotated pronunciation dictionary.

Specifically for TTS, systems often rely on large pronunciation dictionaries to control the phonetic output of generated speech samples. These dictionaries are time-consuming and challenging to acquire, as they require expert knowledge of both target language and phonetic transcription conventions. This is especially relevant for under-resourced or endangered languages. In the absence of phonetic knowledge, TTS models can be trained directly on graphemes [1] or some form of learned representations [2]. Such inputs, however, are problematic for languages with irregular orthography [3] and offer limitations when controlling or correcting the pronunciation of trained and deployed models [3]. Additionally, phoneme-based TTS systems tend to outperform grapheme-based systems, provided that there is enough high-quality annotations, either through handcrafted pronunciation dictionaries or G2P systems [4].

Traditional data-driven approaches to the G2P task used decision trees [5], hidden Markov models [6], joint n-gram models [7], weighted finite-state transducers [8], or neural networks [9]. More recently, various forms of neural networks have been the default approach to G2P conversion, such as LSTMs with or without attention mechanisms [10, 11], or transformer-based architectures [12]. Modern neural network architectures easily outperform traditional data-driven systems [12, 13, 14].

Recent studies, therefore, shifted their focus towards unified multilingual G2P systems that aim to reduce the dissimilarities between languages [14, 15, 16, 17, 18]. Multilingual systems can also benefit under-resourced languages through cross-lingual knowledge transfer. These systems aim to reduce the dependency on large pronunciation dictionaries to quickly scale to new languages [15, 17] or dialects [19]. Alternative approaches to low-resource G2P models use unsupervised text-based pre-training [20], or various forms of text-based data augmentation [21, 22, 23]. Related work for automatic pronunciation learning proposed to acquire knowledge for new languages through audio examples in a zero-shot scenario. This involves learning a new language's phonetic inventory [24], or the automatic labelling of pronunciation using universal phone recognition [25, 26] for downstream tasks such as ASR. Specifically for G2Ps, related work also proposed to leverage audio data to iterate, revise, or complement a G2P system initialized on a small set of annotated materials [27, 28, 29].

In this paper, we propose a method to *improve Grapheme-to-Phoneme models by learning pronunciation examples for out-of-vocabulary words from audio recordings*. Our approach builds on recent studies that use multilingual transformer-based G2P models for cross-lingual knowledge transfer [15, 17], and automatic pronunciation learning from audio using phone recognition [25, 26]. We describe our system in Section 2, while in Section 3 we provide experimental evidence that our systems are scalable and effective in high- and low-resource scenarios.

2. Method

Figure 1 illustrates our approach to improve low-resource G2Ps by automatically learning novel pronunciations from speech recordings. For an unseen target language, we assume that we have a hand-crafted pronunciation dictionary, consisting of <word, pronunciation> pairs, and a speech corpus consisting of <text, audio> pairs, typically used for TTS acoustic modelling. We begin by training a baseline G2P model on the available pronunciation dictionary. We then use the baseline G2P model to hypothesize pronunciations for the vocabulary in the target language speech corpus. We train a Phone Recognizer using <pronunciation, audio> pairs from multilingual data, augmented with <pronunciation, audio> pairs in the target language. We then decode the audio data in the target language. The decode set in the target language may or may not be the same as the train set. Because the phone recognizer operates at the sentence level, we do not have knowledge of word bound-

10.21437/Interspeech.2023-816

aries. The final lexicon learning step aims to align the decoded pronunciation sequences to observed character sequences for word boundary discovery and word-level lexicon learning. In the following sections, we describe the architecture of learnable components of the proposed pipeline.

2.1. Grapheme-to-phoneme conversion

Our approach to G2P conversion is based on a transformer [30] encoder-decoder architecture, implemented with OpenNMT [31]. The encoder and decoder each use 6 layers with 8 self-attention heads. The hidden transformer feed-forward is set to 2048 nodes, while the embedding size is set to 512. We set a dropout rate of 0.1 throughout the model. The model is optimized with `adam`, using the `noam` learning rate scheduler with 8000 warmup steps [30]. The G2P model operates at the word level and inputs a sequence of characters (graphemes), with a prepended language tag. The output phoneme set is defined using the X-SAMPA [32] phonetic alphabet. We pre-train the G2P model on a multilingual pronunciation corpus. We then fine-tune the pre-trained multilingual model for a maximum of 20k steps on available <word, pronunciation> pairs in an unseen target language. For experimental purposes, we ensure that the multilingual model does not have any knowledge of words that occur in unseen languages.

2.2. Phone recognition

Phone recognition aims to annotate speech recordings with the corresponding phoneme sequence. We generate a pronunciation dictionary for the target language "Train Set" using the baseline G2P system. This data is pooled with a large multilingual speech dataset, for which pronunciations are available from large hand-crafted dictionaries. We implement our phone recognition system using Kaldi [33]. We extract Mel Frequency Cepstral Coefficients (MFCCs) from the audio data, which are used to initialize monophone and triphone HMM-GMM models. We then apply Linear Discriminant Analysis (LDA) and a maximum Likelihood Linear Transform (MLLT), which is followed by Speaker Adaptive Training with feature-space MLLR (fMLLR, [34]). We use the features and alignments after this stage to train a time-delay neural network (TDNN, [35]) acoustic model. We do not explicitly provide the acoustic model with language identifiers. The expectation is that the acoustic model will learn to generalize from acoustic realizations across multiple languages, overcoming the potentially noisy labels given by the G2P system for the target language. This is similar to recent approaches to Universal Phone Recognition [25, 26]. When decoding speech samples in the target language, we use a 5gram language model learned on the phone-level generated transcripts for the "Train Set". The language model enforces the system to decode plausible pronunciations restricted to the target language's phonotactics, guided by generated G2P output. The acoustic model aims to hypothesize labels informed by audio samples, guided by universal phonetic knowledge given by multilingual speech examples.

2.3. Lexicon learning

Because we are decoding speech segments at the sentence level, the output of the phone recognition system is a sequence of phonemes without word boundaries. We address this by force-aligning the decoded phoneme sequences to the observed grapheme sequences. We define hidden Markov models (HMMs) for each word in the "Decode Set" vocabulary. The

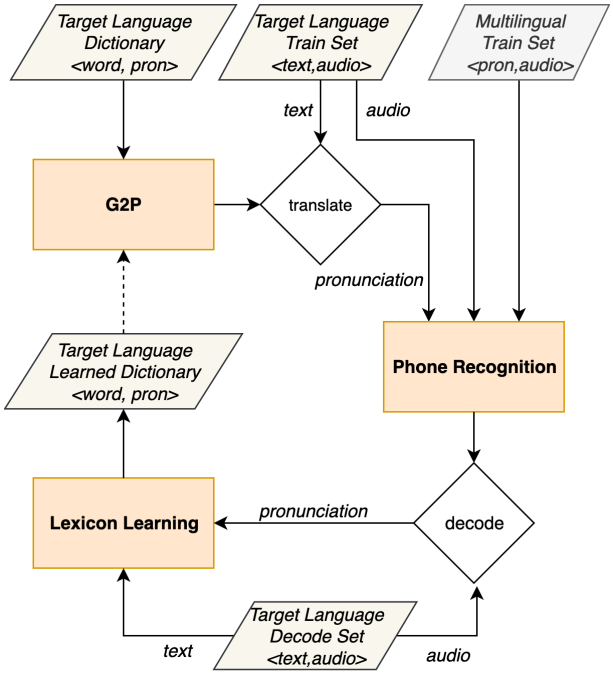

Figure 1: *Approach to pronunciation learning from audio to improve low-resource G2Ps.*

HMMs follow a left-to-right topology with skip connections, with each state corresponding to a grapheme in the word. The probability of each state aligning to a given phoneme is given by a discrete probability distribution over the phoneme space. These distributions are initialized to a uniform distribution and tied across states/graphemes. For each sentence in the "Decode Set", we concatenate the word-level HMMs to form a large sentence-level HMM. Using Viterbi, we find the most likely path through the sentence graph, given the decoded pronunciation sequence. HMM models are implemented and optimized using `pomegranate` [36].

Given hypothesized word boundaries, we can generate learned pronunciation dictionaries by collecting decoded pronunciations for each word in the "Decode Set" vocabulary. Because words can occur multiple times, we define an acceptance threshold k, denoting the number of times we must observe a phoneme sequence paired with any given word. That is, in order to allow a <word, pronunciation> pair to be included in the learned dictionary, we must have decoded the pronunciation for that word independently at least k times. Higher k discards infrequently decoded pronunciations and implies a small learned dictionary of high-frequency words, but higher confidence in the learned pronunciations.

3. Experiments

We evaluate our method on five languages: English, French, Danish, Polish, and Turkish. We begin by training a multilingual G2P model for 1M steps on a training set of 4.5M <word, pronunciation> pairs, pooled across 17 languages. This model is then fine-tuned independently for each of the target languages on a set of seed word pronunciations. We consider initially a low-resource scenario using a seed set of 500 manually-revised words. We select the most frequent 500 words, based on word counts from roughly 2M sentences, extracted from the multi-

Table 1: *Phone Error Rate (PER) for pronunciation dictionaries learned at various thresholds k using a seed set of 500 words (*PER Learned*) and equivalent dictionaries generated by a baseline G2P system (*PER G2P*). Results are presented for the average across 5 languages and separately for English. Columns "Better", "Worse", and "Same" indicate the number of words that are better, worse, or the same with respect to the baseline G2P for the English learned dictionaries.*

	Average (5 languages)		English					
k	PER (Learned)	PER (G2P)	PER (Learned)	PER (G2P)	Num Words	Better	Worse	Same
1	12.99%	**12.53%**	**15.32%**	17.31%	26557	4360 (16.42%)	2431 (9.15%)	19766 (74.43%)
2	**9.27%**	10.25%	**11.13%**	13.45%	12271	1328 (10.82%)	359 (2.93%)	10584 (86.25%)
4	**7.86%**	8.55%	**9.15%**	10.60%	5943	364 (6.12%)	90 (1.51%)	5489 (92.36%)
6	**7.33%**	7.80%	**8.57%**	9.38%	3962	153 (3.86%)	56 (1.41%)	3753 (94.72%)
8	**6.93%**	7.23%	**8.19%**	8.64%	3009	72 (2.39%)	40 (1.33%)	2897 (96.28%)

Table 2: *Phone and Word error rates (PER/WER) for baseline and learned G2P systems at various lexicon learning thresholds k. PER reduction is computed relative to the baseline G2P. Results are averaged across 5 languages.*

System	k	PER	WER	PER Rel. Reduction
Baseline	-	13.02%	50.16%	-
Learned	1	**10.64%**	**45.39%**	**-18.32%**
	2	11.02%	46.06%	-15.36%
	4	11.47%	46.50%	-11.94%
	6	11.69%	47.03%	-10.23%
	8	11.86%	47.53%	-8.89%

Table 3: *Phone Error Rate (PER) for baseline and learned G2P systems at $k = 1$ and $k = 2$ for five languages. Figures in parenthesis indicate the PER reduction relative to the corresponding baseline system.*

Language	Baseline	$k = 1$	$k = 2$
English	19.20%	**16.42%** (-14.48%)	17.01% (-11.38%)
Danish	20.57%	**16.94%** (-17.67%)	17.76% (-13.66%)
French	12.13%	**8.09%** (-33.28%)	8.86% (-26.93%)
Turkish	9.27%	8.44% (-8.90%)	**8.41%** (-9.28%)
Polish	3.95%	3.30% (-16.58%)	**3.07%** (-22.28%)
Average	*13.02%*	***10.64%** (-18.32%)*	*11.02% (-15.37%)*

lingual C4 corpus [37]. We use the baseline fine-tuned G2P system to provide pronunciations for a set of audio recordings in the target language. This "Train/Decode Set" consists of single-speaker studio-quality recordings used for TTS model training. Depending on the language, the amount of available data ranges from 9 to 27 hours. The target-language data is complemented with approximately 165 hours of multilingual speech data, pooled from 6k speakers across 16 languages. The size of the "Multilingual Train Set" differs per language, as we exclude target language data from each iteration. After lexicon learning, we pool the learned pronunciations with the manually-revised seed set and fine-tune the pre-trained multilingual G2P model. We evaluate the G2P systems on test sets consisting of word tokens unseen at training time, and measure results in terms of Phone Error Rate (PER) and Word Error Rate (WER).

3.1. Lexicon learning

We investigate the impact of the threshold k on the learned pronunciation dictionaries. Table 1 shows error rates for the learned dictionaries averaged across the 5 languages, with additional details for English. We include results for equivalent G2P-generated dictionaries, which contain the same words as the corresponding learned dictionaries, but with pronunciations generated by the baseline G2P. As expected, as we increase k, the amount of words allowed in the dictionaries decreases, and so does the overall error rate of the dictionary. When considering the average results, we observe that the dictionaries learned at $k = 1$ underperform when compared with equivalent G2P-generated dictionaries. However, this is not the case for English, where we observe improvements across all values of k. This might be due to amount of available data for the "Train/Decode Sets". The English set has 27 hours of speech data, whereas the underperforming languages at $k = 1$, Polish and Turkish, have 9 and 14 hours, respectively. When considering the distribution of the learned pronunciations (English example in Table 1) , we observe that most decoded pronunciations are the same as those given by the baseline G2P system. However, when the pronunciations are not the same, the phone recognition system has a positive impact over the G2P-generated pronunciations.

3.2. Grapheme-to-Phoneme conversion

We investigate the impact of the learned dictionaries at various thresholds k on the fine-tuned G2P systems. We pool each learned dictionary with the manually-annotated seed set of 500 words and re-train the multilingual G2P system. Table 2 shows results for all values of k, averaged across the 5 languages. We observe a positive impact across all tested values of k, with the best results occurring when $k = 1$. This is an interesting observation, as we have noted that, on average, learned dictionaries at $k = 1$ underperform when compared with corresponding G2P-generated dictionaries (Table 1). These figures suggest that, in a low-resource scenario, the G2P benefits from additional training data, even if containing a higher error rate. In other words, quantity appears to be preferable to quality. We do note also that the G2P likely benefits from the large number of words that were already predicted correctly by the G2P system. Rather than just correcting mistakes from the G2P, the phone recognizer also validates correct pronunciation examples. Table 3 shows results for each of the evaluated languages at $k = 1$ and $k = 2$. Turkish and Polish achieve the best results at $k = 2$. We hypothesize that this might be due to the already low error rates of the G2P for these languages. Turkish and Polish have more straightforward grapheme-to-phoneme correspondence, when compared with the other languages evaluated. It might be that, for such cases, higher thresholds are more suitable, leading to smaller, but higher quality dictionaries.

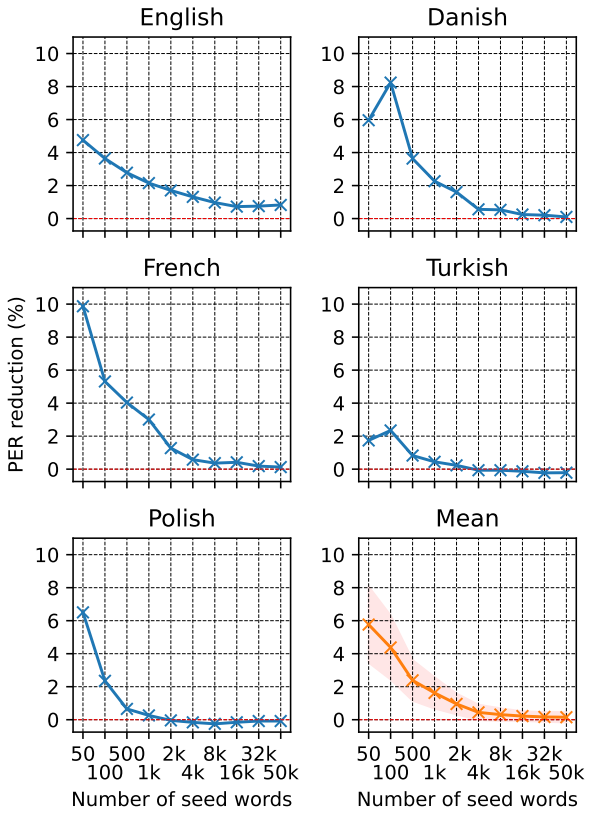

Figure 2: *Phone Error Rate (PER) reduction between baseline and learned G2P system across varying number of seed words. Results are presented for 5 languages and their mean.*

3.3. Amount of seed data

Thus far, we have presented results for low-resource G2Ps using a seed set of 500 annotated words. We investigate the behaviour of our approach as we vary the amount of seed data. The pipeline is identical to that described in Section 2, and we change only the size of the manually-annotated pronunciation dictionaries for the target language. We define the dictionaries such that each incremental word set is a superset of the one immediately before. Figure 2 illustates results for the 5 languages and their average. For simplicity, we visualize the absolute PER difference between the baseline and the learned system at $k = 1$. We observe that our method has the highest impact on low-resource scenarios, when 1000 words or less are available. For these systems, we reduce the PER by 2-6%, on average across all languages. The biggest improvements occur with French, Danish and Polish when only with 50 annotated words are available. Our approach reduces PER by 6-10% absolute over the baseline system. For medium and high-resource scenarios (more than 2000 words), results show variable improvements across languages. Languages such as Polish and Turkish observe a deterioration of PER over the baseline, whereas languages such as English, Danish or French observe an improvement. These changes, however, are small (~1% PER).

3.4. Iterative self-training

Given the promising results after a single pass of our pipeline, we investigate additional improvements as we iterate in a type of "self-training" scenario. For each iteration, we choose a G2P checkpoint based on the phone error rate measured on a small validation set. This checkpoint is then used to annotate the audio data for phone recognition. All systems are trained from scratch for each iteration using the same audio data and a constant $k = 1$. This is a slightly different setup from earlier experiments, which is why baseline figures differ. Results are presented in Table 4 for the evaluation sets. Although the impact of the learned pronunciation dictionaries decreases as we iterate, we always observe improvements by self-training. For example, in the English system with a seed set of 100 words, we reduce PER relative to the baseline from 17.14% (iteration 1) to 20.96% (iteration 5).

Table 4: *Phone Error Rate (PER) over multiple iterations of pronunciation learning for English and Danish using a seed set of 100 and 500 words at $k = 1$. The columns "Rel. Red." denote the PER reduction relative to the previous iteration.*

Language	System	100 seed words		500 seed words	
		PER	Rel. Red.	PER	Rel. Red.
English	Baseline	22.26%	-	22.02%	-
	Iter 1	18.44%	-17.14%	17.62%	-11.97%
	Iter 2	17.71%	-3.96%	17.16%	-2.61%
	Iter 3	17.71%	0.00%	16.97%	-1.14%
	Iter 4	17.69%	-0.14%	**16.87%**	-0.59%
	Iter 5	**17.59%**	-0.54%	17.03%	0.95%
Danish	Baseline	39.60%	-	20.63%	-
	Iter 1	30.56%	-22.81%	**17.29%**	-16.19%
	Iter 2	30.26%	-1.00%	17.31%	0.12%
	Iter 3	29.81%	-1.50%	17.35%	0.23%
	Iter 4	**29.56%**	-0.82%	17.46%	0.63%
	Iter 5	29.56%	0.00%	17.43%	-0.20%

4. Conclusion and future work

We have proposed an approach to improve grapheme-to-phoneme models by learning pronunciations from speech recordings. We showed evidence that our method consistently improves G2P systems for low-resource scenarios across 5 different languages. Results indicate that pronunciation dictionaries learned with a low threshold lead to the best results. This suggests that the quantity of words tends to be preferred over their quality. The impact of the pronunciation learning system is higher for low-resource scenarios, when the G2P system is more prone to errors. We additionally observe that iterating over the pipeline leads to increase performance for the G2P system.

For future work, we aim to investigate methods for automatic pronunciation learning in high-resource scenarios. Our results suggest that the impact of our approach is small when a large amount of data is available. We aim to investigate solutions to learn novel pronunciations for infrequent words with irregular grapheme-to-phoneme correspondence, such as proper names, loan words, or domain-specific tokens. Our experimental evidence relied on target language audio data that was restricted to a single-speaker set of recordings. Further work should explore the impact of the type or amount of audio data in the target language. Using larger multi-speaker speech recordings in the target language might lead to improved results, particularly for the low-resource scenarios. Additionally, we might achieve improved performance with more robust speech recognition systems, perhaps pre-trained in a self-supervised fashion.

5. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *Proc. ICASSP*, 2018.

[2] H. Siuzdak, P. Dura, P. van Rijn, and N. Jacoby, "WavThruVec: Latent speech representation as intermediate features for neural speech synthesis," *Proc. Interspeech*, 2022.

[3] J. Taylor and K. Richmond, "Analysis of pronunciation learning in end-to-end speech synthesis." *Proc. Interspeech*, 2019.

[4] J. Fong, J. Taylor, K. Richmond, and S. King, "A comparison between letters and phones as input to sequence-to-sequence models for speech synthesis," in *10th ISCA Speech Synthesis Workshop*, 2019, pp. 223–227.

[5] A. W. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *The third ESCA/COCOSDA workshop (ETRW) on speech synthesis*, 1998.

[6] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[7] L. Galescu and J. F. Allen, "Pronunciation of proper names with a joint n-gram model for bi-directional grapheme-to-phoneme conversion," in *Seventh International Conference on Spoken Language Processing*, 2002.

[8] J. R. Novak, N. Minematsu, and K. Hirose, "WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding," in *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, pp. 45–49.

[9] R. Damper, Y. Marchand, M. Adamson, and K. Gustafson, "A comparison of letter-to-sound conversion techniques for English text-to-speech synthesis," *Proceedings of the Institute of Acoustics*, vol. 20, no. 6, pp. 245–254, 1998.

[10] S. Toshniwal and K. Livescu, "Jointly learning to align and convert graphemes to phonemes with neural attention models," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 76–82.

[11] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Proc. ICASSP*, 2015.

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

[13] B. Milde, C. Schmidt, and J. Köhler, "Multitask sequence-to-sequence models for grapheme-to-phoneme conversion." *Proc. Interspeech*, 2017.

[14] B. Peters, J. Dehdari, and J. van Genabith, "Massively multilingual neural grapheme-to-phoneme conversion," *Proc. First Workshop on Building Linguistically Generalizable NLP Systems*, 2017.

[15] A. Sokolov, T. Rohlin, and A. Rastrow, "Neural machine translation for multilingual grapheme-to-phoneme conversion," *Proc. Interspeech*, 2019.

[16] O. ElSaadany and B. Suter, "Grapheme-to-phoneme conversion with a multilingual transformer model," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2020, pp. 85–89.

[17] M. Yu, H. D. Nguyen, A. Sokolov, J. Lepird, K. M. Sathyendra, S. Choudhary, A. Mouchtaris, and S. Kunzmann, "Multilingual grapheme-to-phoneme conversion with byte representation," *Proc. ICASSP*, 2020.

[18] J. Zhu, C. Zhang, and D. Jurgens, "ByT5 model for massively multilingual grapheme-to-phoneme conversion," *Proc. Interspeech*, 2022.

[19] E. Engelhart, M. Elyasi, and G. Bharaj, "Grapheme-to-phoneme transformer model for transfer learning dialects," *arXiv preprint arXiv:2104.04091*, 2021.

[20] L. Dong, Z.-Q. Guo, C.-H. Tan, Y.-J. Hu, Y. Jiang, and Z.-H. Ling, "Neural grapheme-to-phoneme conversion with pre-trained grapheme models," *Proc. ICASSP*, 2022.

[21] X. Yu, N. T. Vu, and J. Kuhn, "Ensemble self-training for low-resource languages: Grapheme-to-phoneme conversion and morphological inflection," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2020, pp. 70–78.

[22] B. Hauer, A. A. Habibi, Y. Luan, A. Mallik, and G. Kondrak, "Low-resource G2P and P2G conversion with synthetic training data," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2020, pp. 117–122.

[23] M. Hammond, "Data augmentation for low-resource grapheme-to-phoneme mapping," in *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2021, pp. 126–130.

[24] P. Żelasko, S. Feng, L. M. Velázquez, A. Abavisani, S. Bhati, O. Scharenborg, M. Hasegawa-Johnson, and N. Dehak, "Discovering phonetic inventories with crosslingual automatic speech recognition," *Computer Speech & Language*, vol. 74, p. 101358, 2022.

[25] 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 *Proc. ICASSP*, 2020.

[26] O. Klejch, E. Wallington, and P. Bell, "Deciphering speech: a zero-resource approach to cross-lingual transfer in ASR," *Proc. Interspeech*, 2021.

[27] N. Goel, S. Thomas, M. Agarwal, P. Akyazi, L. Burget, K. Feng, A. Ghoshal, O. Glembek, M. Karafiát, D. Povey *et al.*, "Approaches to automatic lexicon learning with limited training examples," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 5094–5097.

[28] J. Route, S. Hillis, I. C. Etinger, H. Zhang, and A. W. Black, "Multimodal, multilingual grapheme-to-phoneme conversion for low-resource languages," in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 2019, pp. 192–201.

[29] A. Aquino, J. L. Tsang, C. R. Lucas, and F. de Leon, "G2P and ASR techniques for low-resource phonetic transcription of tagalog, cebuano, and hiligaynon," in *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*. IEEE, 2019, pp. 1–5.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017, pp. 67–72.

[32] J. C. Wells, "Computer-coding the IPA: a proposed extension of SAMPA," *Revised draft*, vol. 4, no. 28, p. 1995, 1995.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[34] S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved feature processing for deep neural networks." *Proc. Interspeech*, 2013.

[35] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," *Proc. Interspeech*, 2015.

[36] J. Schreiber, "Pomegranate: fast and flexible probabilistic modeling in Python," *Journal of Machine Learning Research*, vol. 18, no. 164, pp. 1–6, 2018.

[37] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proc. NAACL*, 2021, pp. 483–498.