Cross-lingual transfer of phonological features for low-resource speech synthesis

Cross-lingual Transfer of Phonological Features for Low-resource Speech Synthesis

Dan Wells, Korin Richmond

The Centre for Speech Technology Research,
University of Edinburgh, United Kingdom
{dan.wells, korin.richmond}@ed.ac.uk

Abstract

Previous work on cross-lingual transfer learning in text-to-speech has shown the effectiveness of fine-tuning phonemic representations on small amounts of target language data. In other contexts, phonological features (PFs) have been suggested as a more suitable input representation than phonemes for sharing acoustic information between languages, for example in multilingual model training or for code-switching synthesis where an utterance may contain words from multiple languages. Starting from a model trained on 14 hours of English, we find that cross-lingual fine-tuning with 15 minutes of German data can produce speech with subjective naturalness ratings comparable to a model trained from scratch on 4 hours of German, using either phonemes or PFs. We also find a modest but statistically significant improvement in naturalness ratings using PFs over phonemes when training from scratch on 4 hours of German.

Index Terms: speech synthesis, low-resource, cross-lingual, transfer learning

1. Introduction

Phonemes are often used as atomic input symbols to text-to-speech (TTS) systems as an explicit representation of the pronunciation of input text [1]. This is useful even for large neural sequence-to-sequence models which have the capacity to learn implicit pronunciation models directly from text inputs but which may make mistakes compared to grapheme-to-phoneme (g2p) conversion models trained on high-quality lexicons [2, 3]. Such large TTS models are typically trained using tens of hours of audio data with associated text transcriptions, which alongside the specialist linguistic knowledge required to convert raw text into phoneme strings are expensive resources to attain and limit the application of these models to a small proportion of the world's 7,000 languages.

For languages with minimal data resources for TTS model training, we might instead consider fine-tuning an existing model from another language with much more data available. In a phoneme-based system, input embeddings for phonemes common to both languages may be initialised in the target-language model by copying source-language parameters directly. For phonemes unique to the target language, however, some additional method is required to determine whether any particular source phoneme may provide a suitable starting point. In [4], a learned mapping is compared to a unified symbol space constructed by aligning phoneme symbols in each language using linguistic expertise, with both approaches achieving similar naturalness ratings when initialised from a model of 24 hours of English speech and fine-tuned with 15 minutes of Mandarin data. These fine-tuning approaches outperform a baseline with random initialisation of Mandarin phoneme embeddings.While the learned phoneme mappings were found largely to correspond with expert mappings, some target-language phonemes went unmapped due to low confidence in the suggested source phoneme and still had to be initialised from scratch. This follows from the atomic nature of phonemic input symbols, such that automatic phoneme mapping is an all-or-nothing approach.

An alternative approach is to decompose phonemic symbols into sets of distinctive phonological features (PFs) corresponding to articulatory attributes such as tongue position, degree of closure and voicing [5]. This representation reveals shared characteristics between phonemes which are not evident when considering only their atomic symbols in a transcription system such as the International Phonetic Alphabet (IPA) [6], and makes it possible to transfer learned embeddings for individual features between languages and so compose representations for target-language phonemes completely unseen during source model training. Previous work has used PF representations to share acoustic information between languages during multilingual model training for LSTM-RNN [7, 8] or feed-forward [9] neural network acoustic models. These models typically include PFs as part of a wider set of linguistic features, sometimes including phoneme labels as well, drawn from a unified symbol space across all training languages. In [7], for example, this data pooling approach using PFs was found to improve naturalness ratings for low-resource languages relative to individual voices trained using only data from those languages.

Our work is closest to that of [10], who use PFs in an encoder-decoder model with attention based on [11] to enable zero-shot synthesis of code-switched speech. They showed that a model trained on one language can be used to generate intelligible speech in a completely unseen target language with no acoustic training data available. Although they evaluated their system in an extreme setting with entire utterances comprised of target-language words, the work was motivated by the need to handle individual vocabulary items being embedded within source-language utterances, for example foreign names. We are directly interested in synthesising full utterances in the target language, and so apply a similar method in a transfer learning context, starting from a high-resource English source model and fine-tuning with either 15 minutes or 4 hours of transcribed German data. Also similar to [10], we rely on considerable lexical resources for g2p conversion prior to PF expansion, so that 'low-resource' in our case refers primarily to this relatively limited availability of transcribed speech data.

2. Phonological features

We use a set of binary phonological features derived from those introduced in Chomsky and Halle's *Sound Pattern of English* (SPE) [5]. In this formalism, each phoneme is represented as a

binary vector of 24 features as listed in Table 1. Of these, 19 are a selection of SPE features which adequately describe the phonetic inventories of English and German, and are essentially phonological in nature. We also add 5 features to capture aspects of input text strings, for example representing the end of a sentence or other prosodically-relevant punctuation types.

Table 1: *SPE-style phonological features.*

Category	Features
Major class	syllabic, consonantal, sonorant
Cavity	coronal, anterior, high, low, front, back, round, nasal, lateral, constricted glottis
Manner	continuant, tense, delayed release
Source	voice, strident, subglottal pressure
Text	space, end of sentence, question, exclamation, other punctuation

Following discussion in [5, pp. 353–355] on the treatment of glides relative to high vowels, e.g. /j/ vs. /i/, and to account for syllabic consonants, e.g. /n̩/ in 'button', we replace the original SPE *vocalic* feature with *syllabic*. We also add an explicit *front* feature for horizontal tongue body position alongside *back* to allow for distinction of central vowels in our feature system, e.g. open-mid front /ɛ/ [+*front*, −*back*] vs. central /ɜ/ [−*front*, −*back*]. All other features and mappings between phonetic segments and phonological feature vectors follow closely with those laid out in [5].

For a concrete example, consider our scenario of fine-tuning a high-resource English model using a small amount of German data. Both languages' phonemic inventories include an unvoiced velar plosive /k/, while only German natively makes use of an unvoiced velar fricative /x/. These two sounds share many features, both being produced at the same place of articulation in the mouth with the back of the tongue raised and without vibration of the vocal folds. The main difference between the two is the degree of closure in the oral cavity, with transient but complete interruption of airflow in the case of /k/ compared to narrowing of the vocal tract enough to generate turbulent airflow and constant noise for /x/. If we were only to consider the atomic symbols /k/ and /x/, for example by converting them to one-hot indices in a neural embedding table, these similarities may not be apparent, and we would have to make hard decisions about a possible mapping between these sounds if we wanted to transfer acoustic information learned on English data to our German model, as in [4]. In our PF representation, on the other hand, these two phonemes differ only in the specification of the feature *continuant*, which is − for /k/ and + for /x/. As such, at the beginning of our fine-tuning regime the encoder of our German model is initialised with a representation of /x/ which already contains much information learned from the English /k/, supplemented by [+*continuant*] English phonemes such as /s/. Although we do not test it formally here, we find these initial representations to produce somewhat intelligible German speech even before any target-language data has been seen by the model, as in [10], albeit retaining our English source speaker's vocal quality and accent.

Our binary feature representation largely overlaps with that used in PanPhon [12], and differs from the multi-valued features used in [10], which map more directly to IPA categories such as *vowel frontness* or *consonant place*. While our feature set gives a more compact representation, with 24 features vs. 60 in [10] (after conversion to binary vectors), it is perhaps less interpretable in familiar linguistic terms, for example with the *palatal* place of articulation feature in a multi-valued representation instead being composed from [+*high*, −*low*, −*back*] feature specifications in our system. Previous work on phonological feature detection from speech [13] found similar performance between an SPE-style binary feature system like ours and multi-valued features, and [8] showed improvements for multilingual TTS training using inputs augmented with PFs of both kinds, suggesting that either formalism may be adequate for speech processing tasks.

3. Methodology

3.1. Speech data

For our English voice we use part of the M-AILABS Speech Dataset [14], from the female US speaker *mary_ann*. We only use recordings from the *northandsouth* text, as other recordings from this speaker have a slight reverberant quality. For German, we use the CSS10 dataset [15], which provides a single female speaker. Both corpora are drawn from non-professional audiobook recordings made as part of the LibriVox project [16].

The CSS10 German corpus comprises 16 hours of speech sampled at 22.05 kHz, whereas M-AILABS provides 18 hours sampled at 16 kHz. For our English source models we randomly sample 14 hours (hereafter labelled 840 minutes) from M-AILABS as a training set and 90 minutes for validation. For German, we sample training sets of 15 minutes and 4 hours (240 minutes) and validation sets of 5 and 20 minutes respectively to match the low-resource training setting [17]. A disjoint set of 70 utterances is held out to synthesise listening test stimuli. All German utterances are downsampled to 16 kHz to match the English data. Table 2 summarises these data partitions.

Table 2: *Dataset summary: total number of utterances, average length in phonemes and average duration in seconds.*

Dataset	Utterances	Phones	Duration
EN-train-840	6975	97	7.23
EN-val-90	754	98	7.16
DE-train-240	1698	102	8.48
DE-val-20	153	94	7.85
DE-train-15	103	106	8.76
DE-val-5	38	98	8.12
DE-test	70	87	7.51

As part of dataset selection, we exclude from the English data any utterances with raw text transcriptions longer than 200 characters, and from the German any transcripts longer than 170 characters. This only serves to remove outliers from each dataset, and does not affect the overall distribution of observed transcript lengths. We also exclude any utterances from M-AILABS with digits in their raw transcripts, since we found the normalised transcripts provided did not match the words spoken in several instances. For German test utterances, we select only those with transcripts ending in some kind of intonational phrase-final punctuation $p \in \{.!?;\}$. We do this to increase the proportion of test stimuli which correspond to complete sentences, given that the CSS10 corpus was created by automatically segmenting long audiobook chapters and is not guaranteed

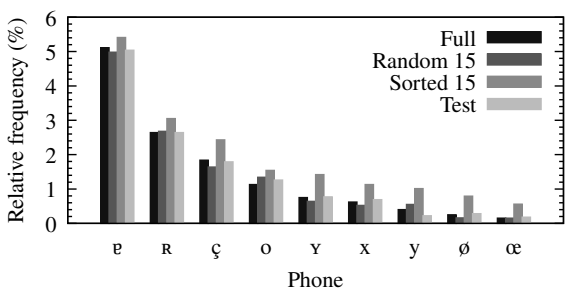


Figure 1: *German-specific phoneme frequencies for different subsets of CSS10 data.*

to have a one-to-one correspondence between segmented utterances and source text sentences.

When sampling German training subsets, we first sort utterances by how many phonemes they contain which are specific to German and therefore unseen during English source model training. We then select utterances starting with the most unseen phoneme types (out of 9 total) until the target dataset duration is met, so maximising training examples for these phones in our low-resource setting. We consider this a valid approach when some lexical resources are available in the target language, since prompt selection in this way can be done before recording any audio. Sorting by unseen phoneme type counts tends to give a greater increase in relative frequency of the least frequent phones, whereas sorting by token counts instead boosts the most frequent unseen phones. Figure 1 shows the effect of this procedure when sampling 15 minutes of German audio; the effect is reduced for 240 minute subsets, as even random sampling begins to exhaust the supply of the least frequent phones in the data. Type counts also restrain the tendency to select longer utterances compared to token counts, although as seen in Table 2 German training utterances are still slightly longer on average than validation utterances, which are not sorted by unseen phoneme counts before sampling.

3.2. Grapheme-to-phoneme conversion

To encode inputs using phonological features, we first need to convert input text to IPA phoneme strings. Where possible, we look up pronunciations in a lexicon: the General American surface form of Combilex [18] for English and the German lexicon from MaryTTS [19], mapping their individual phone sets to IPA symbols. To handle out-of-vocabulary items in each language we train g2p models from these lexicons using the Phonetisaurus toolkit [20].

3.3. Model details

We use a modified Tacotron 2 [11] architecture to predict acoustic features from text, based on the Mozilla TTS implementation [21]. Following [10], in our PF-based models we replace phoneme embeddings with a single linear layer over binary feature inputs, with matching 512-dimensional hidden representations. Mozilla TTS retains the reduction factor used in the original Tacotron [22], predicting r output frames per decoder step. We had better results training our English source model with $r = 2$, predicting frames in pairs rather than individually as in [11], and use the same reduction factor when fine-tuning German models. All other architectural details match [11].

We train English source models for 100k steps, using a Rectified Adam optimiser [23] with batch size 32 and learning rate 1×10^{-4}. German-only models use the same training hyperparameters but run for 60k steps, and fine-tuned models run for 60k steps with a learning rate of 3×10^{-5}. In this way, all German models using the same data split have equal exposure to training examples in that language, and we can evaluate the potential of each model and training scheme in matched data settings. As 240 minutes of speech is much less than is typically used to train sequence-to-sequence neural TTS models such as ours, we were concerned to ensure that our German-only models were adequately trained for fair comparison with the fine-tuned models which also see 14 hours of English data. The cutoff at 60k training steps was chosen to enable strong alignments between input and output timesteps to be learned by the German-only models, which we found to be the major factor preventing gross synthesis errors for those systems.

When fine-tuning phonological feature-based models, which we label *F-{15,240}-ft* depending on amount of German data used, all model parameters are copied directly from the English source model, since PF inputs are completely shared between the two languages. For phoneme-based models (*P-{15,240}-ft*), we copy learned English embeddings directly for all shared phonemes. For German-specific phonemes, we follow [10] and initialise their embeddings with the closest English phoneme largely according to PF specifications. This presents a stronger baseline to test PF systems against compared to leaving them with untouched random initialisations from the English pre-training stage. Figure 3(b) indicates the English phonemes selected to initialise German-specific phoneme embeddings.

We found that stop token prediction did not fare well when transferring from English to German. Fine-tuning this component led to 69% of synthesised utterances from *240-ft* systems and 17% from *15-ft* hitting an upper limit on decoder steps, often producing audible 'babbling' for the additional duration following synthesis of text prompts. This may be caused by mismatches in utterance-final prosody or other acoustic differences between English and German, or perhaps the increased proportion of sentence-fragment utterances in the German data compared to English. Models trained from scratch on our German data didn't exhibit this issue to the same degree, and re-initialising stop token projection weights rather than transferring from English source parameters during fine-tuning largely addresses the problem. Synthesis of our final *15-ft* test stimuli saw no utterances reaching the maximum decoder steps, while the proportion in *240-ft* systems was reduced to 17%.

We also train a Parallel WaveGAN vocoder [24] on our English dataset to generate audio from predicted acoustic features (implementation based on [25]). This model is trained as described in [24], for 400k training steps. We find the vocoder to transfer well to the unseen speaker in our German data without additional fine-tuning (cf. discussion in [26]), though since vocoder training requires only audio and extracted acoustic features and not aligned text transcripts, target-language vocoder training could be viable even in a low-resource setting.

3.4. Listening tests

We evaluate system performance by conducting MUSHRA-style listening tests [27]. Each test panel comprises multiple versions of the same utterance synthesised by each system under test, plus a natural speech reference (recorded by the same speaker used in training) and vocoded speech using mel spectral features extracted from the reference (copy synthesis). Natural speech is presented as an explicit reference and also included as

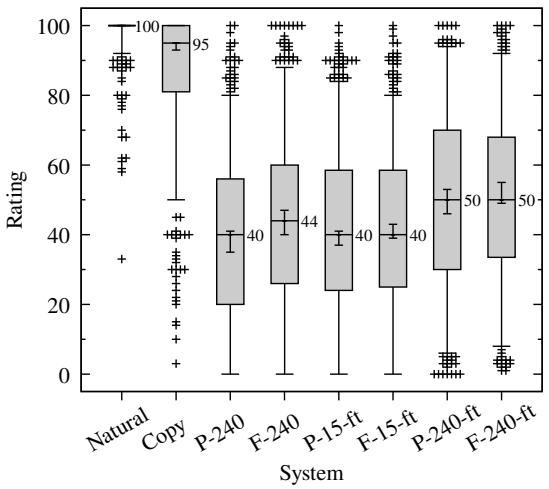


Figure 2: *MUSHRA naturalness ratings per system. Central bars indicate median ratings with 99% confidence intervals, boxes span 25–75% quartiles and whiskers cover 95% of results for each system. Outliers are marked with +.*

a hidden reference among other test samples, randomly ordered. Given the difficulty in identifying a suitable 'anchor' stimulus to serve as a lower bound for expected quality in speech synthesis, no such stimulus is included in our tests; each panel therefore contains 9 audio samples in total. Participants are asked to listen to the reference and then to provide a rating from 0–100 for each test sample reflecting how 'natural' they sound compared to the reference. To proceed to the next panel, at least one sample must be rated at 100 on the naturalness scale.

We recruited 40 participants through Prolific, filtering for native speakers of German, and conducted listening tests on the Qualtrics survey platform. Each participant completed 16 panels randomly allocated from our held-out set of 70 test utterances, with each utterance being rated by 9 or 10 participants in total. The average test duration was 35 minutes, and participants were paid £5 for their time.

4. Results

4.1. Subjective evaluation

The MUSHRA naturalness ratings for each system gathered through our subjective listening tests are shown in Figure 2. All systems present a wide range of participant ratings, including copy synthesis and even the hidden reference natural speech to some extent. We did not find any systematic source for this (e.g. particular stimuli or participants), and attribute it to natural variation in subjective ratings. Audio samples of test stimuli are available online.[1]

We test for significant differences between systems using double-sided pairwise Wilcoxon signed-rank comparisons, applying the Bonferroni correction with $\alpha = 0.01$ (for 28 pairwise comparisons, significance is found at $p < 0.00036$). Both *F-240-ft* and *P-240-ft* are significantly more natural than all other TTS systems, but there is no significant difference between them. The two systems fine-tuned with 15 minutes of German data are not significantly different from each other or either of the two systems trained on 240 minutes of German data only. The German-only system trained with PF inputs (*F-240*) is significantly more natural than the equivalent system using phonemes (*P-240*).

From these results, we see that by fine-tuning a source model trained on a high-resource language with as little as 15 minutes of annotated speech data in the target language, it is possible to match peformance against a system trained on 240 minutes of data from the target language alone. Furthermore, significant improvements in naturalness of the synthesised voice can be found by increasing the amount of fine-tuning data to 240 minutes. This is true for both phoneme- and PF-based systems, confirming previous results on fine-tuning from phoneme inputs in [4] and effectively extending the method to PFs with their more flexible and straightforward method for initialising target-language encoder representations compared to atomic phoneme mappings. We also find that, in the absence of a source model in another language, PFs can give a significant boost to naturalness ratings compared to phonemes in a low-resource setting with 240 minutes of target-language data.

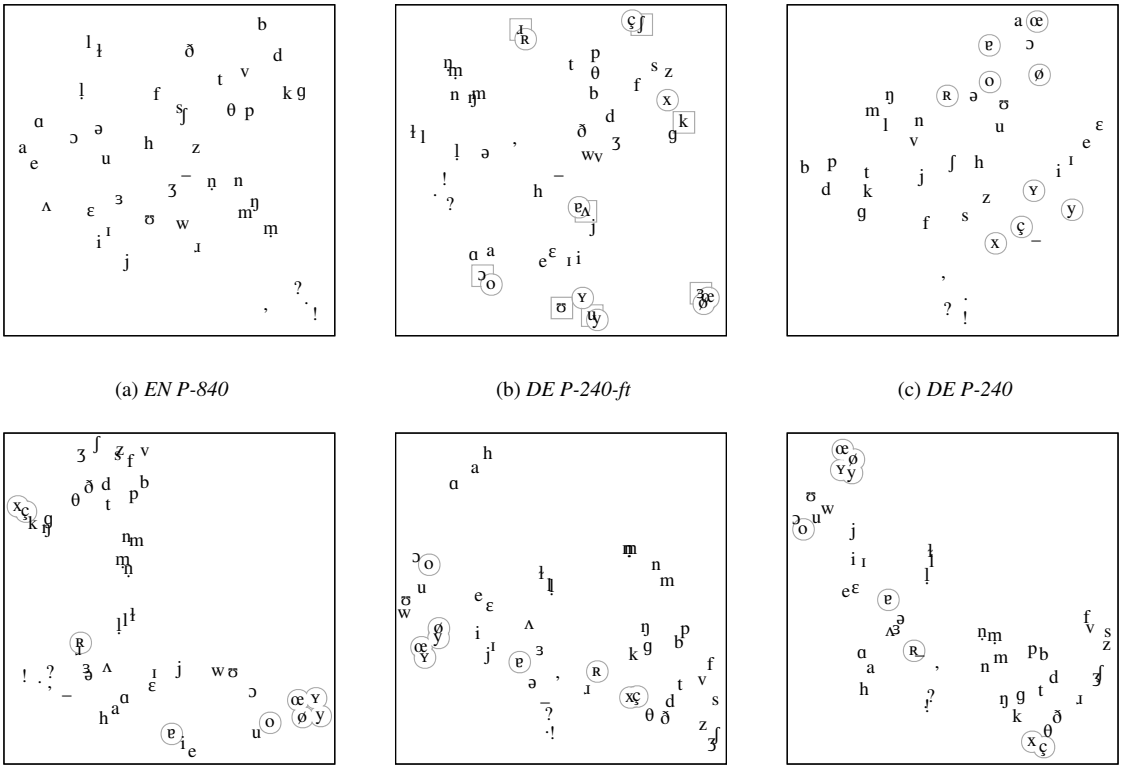
4.2. Input embeddings

To analyse the learned representations of phonemes in our models, we project input embeddings to two dimensions using UMAP [28], as shown in Figure 3. We encourage somewhat more local structure in our projections by reducing the default number of neighbouring points considered in the reference implementation of UMAP from 15 to 5, based on the intuition that individual phonemes are typically more closely related to a small subset of other sounds in any particular phoneme inventory in which they may be found. For clarity in Figures 3(a) *EN P-840* and 3(c) *DE P-240*, we exclude the randomly-initialised embeddings of phonemes from the other language (which are never updated during training for these systems) when projecting the embedding spaces. Although UMAP is a stochastic algorithm, we found the projections of our learned embeddings to be quite consistent across multiple runs.

There is some apparent structure for both phoneme and PF representations, with vowels and consonants grouped separately, distinct consonant classes grouped together (plosives, fricatives and nasals especially) and voiced and unvoiced consonants at the same place of articulation lying close together. Some higher-level relationships appear important for PF projections, for example with vowels seemingly arrayed primarily along an axis of rounding and within those [$\pm$*round*] clusters by frontness and height. For consonants, the *back* feature also appears to be significant above manner of articulation, with [+*back*] plosives /k/ and /g/, fricatives /ç/ and /x/ and the nasal /ŋ/ tending to be separated from their anterior counterparts.

Interesting differences may be seen in the behaviour of the two German-specific fricatives, velar /x/ and palatal /ç/, between the *P-240* model trained only on German data and *P-240-ft* which was fine-tuned from English phoneme representations. In *P-240*, these sounds are grouped closely together with other fricatives, and are quite apart from any plosive consonants. In the fine-tuned model, on the other hand, the separation between fricative and plosive is less clear, specifically with velar plosives /k/ and /g/ appearing close to /x/, while /ç/ is somewhat separated from the other fricatives along with /ʃ/. Notably, these two phonemes were initialised from the learned English embeddings for /k/ and /ʃ/, respectively. If we consider other German-specific phonemes and the corresponding English phonemes from which they were initialised, there is apparently very little movement from the English starting points in all cases. This

[1] https://dan-wells.github.io/pf-tts



(a) *EN P-840* (b) *DE P-240-ft* (c) *DE P-240*

(d) *EN F-840* (e) *DE F-240-ft* (f) *DE F-240*

Figure 3: *UMAP projections of input symbol embeddings for English and German models using phonemes (a–c) and PFs (d–f). German-specific phonemes are marked by circular outlines, and English phonemes used to initialise their representations in (b) by squares. Unseen German phonemes are included in (d) to show that novel combinations of PFs also produce sensible representations.*

could be a result of the high-dimensional (512) phoneme embedding space used: in such a large representational space, it may be possible to adapt a plosive /k/ to sound adequately like its corresponding fricative /x/ by making small perturbations in many dimensions. This high-dimensional perturbation might not then be preserved during low-dimensional projection as we have done here. By comparison, these phonemes pattern consistently across both *DE* models trained from scratch and through fine-tuning when using PFs, as well as in *EN F-840*, where they were completely unseen during training. This supports the notion that PFs should be a stable representation cross-linguistically, backing up observed improvements in multilingual training contexts in [7, 8].

5. Conclusion

In this work, we experimented with phonological feature vector inputs to TTS models in a transfer learning context. We confirmed previous results which showed that cross-lingual fine-tuning is a viable method for training synthetic voices with limited amounts of target language data, with source models trained on 14 hours of English being adapted using 15 minutes of German data matching the subjective naturalness ratings of models trained from scratch using 4 hours of German data only. We found this result to hold for PFs as well as phonemes, but consider PFs to bring practical benefits with regard to ease of parameter sharing in this transfer learning context. We also found a small but statistically significant improvement in naturalness ratings when training a voice from scratch on 4 hours of German data using PFs over phonemes.

While the models trained here may be called 'low-resource' in terms of annotated speech data available in the target language, we still rely on considerable lexical resources for grapheme-to-phoneme conversion of input text before we can expand IPA symbols to PFs. Future work may consider the application of recent approaches to multilingual g2p systems [29] as part of this low-resource pipeline, or make use of additional pre-existing linguistic resources such as the PHOIBLE phonological inventory database [30]. Following our analysis of learned input embeddings, we would also like to investigate more constrained embedding spaces to encourage more efficient parameter sharing, especially for phonemes which remain a common choice of input representation for TTS.

6. Acknowledgements

We would like to thank Gustav Eje Henter for helpful discussion on analysing MUSHRA results and Alexander Schotthöfer for translating experimental materials into German. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences.

7. References

[1] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation Mixing for TTS Synthesis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5906–5910.

[2] J. Taylor and K. Richmond, "Analysis of Pronunciation Learning in End-to-End Speech Synthesis," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 2070–2074.

[3] J. Fong, J. Taylor, K. Richmond, and S. King, "A Comparison of Letters and Phones as Input to Sequence-to-Sequence Models for Speech Synthesis," in *10th ISCA Speech Synthesis Workshop*. ISCA, Sep. 2019, pp. 223–227.

[4] Y.-J. Chen, T. Tu, C.-c. Yeh, and H.-Y. Lee, "End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 2075–2079.

[5] N. Chomsky and M. Halle, *The Sound Pattern of English*. New York: Harper & Row, 1968.

[6] International Phonetic Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[7] A. Gutkin, "Uniform Multilingual Multi-Speaker Acoustic Model for Statistical Parametric Speech Synthesis of Low-Resourced Languages," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 2183–2187.

[8] A. Gutkin, M. Jansche, and T. Merkulova, "FonBund: A Library for Combining Cross-lingual Phonological Segment Data," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[9] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker Adaptation of a Multilingual Acoustic Model for Cross-Language Synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7629–7633.

[10] M. Staib, T. H. Teh, A. Torresquintero, D. S. R. Mohan, L. Foglianti, R. Lenain, and J. Gao, "Phonological Features for 0-Shot Multilingual Speech Synthesis," in *Interspeech 2020*. ISCA, 2020, pp. 2942–2946.

[11] 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. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on Mel Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 4779–4783.

[12] D. R. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. Levin, "PanPhon: A Resource for Mapping IPA Segments to Articulatory Feature Vectors," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 3475–3484.

[13] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 333–353, Oct. 2000.

[14] Munich Artificial Intelligence Laboratories GmbH, "The M-AILABS Speech Dataset," 2019. [Online]. Available: https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/

[15] K. Park and T. Mulc, "CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages," in *Interspeech 2019*. ISCA, 2019, pp. 1566–1570.

[16] LibriVox, "LibriVox – Free public domain audiobooks." [Online]. Available: https://librivox.org/

[17] K. Kann, K. Cho, and S. R. Bowman, "Towards Realistic Practices In Low-Resource Natural Language Processing: The Development Set," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3340–3347.

[18] K. Richmond, R. A. J. Clark, and S. Fitt, "Robust LTS Rules with the Combilex Speech Technology Lexicon," in *Interspeech 2009*, 2009, pp. 1295–1298.

[19] M. Schröder and J. Trouvain, "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching," in *4th ISCA Speech Synthesis Workshop*, 2001.

[20] J. R. Novak, N. Minematsu, and K. Hirose, "Failure Transitions for Joint n-Gram Models and G2P Conversion," in *Interspeech 2013*, 2013, pp. 1821–1825.

[21] "Mozilla/TTS," Mozilla, Apr. 2021. [Online]. Available: https://github.com/mozilla/TTS

[22] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 4006–4010.

[23] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," in *Eighth International Conference on Learning Representations (ICLR 2020)*, Apr. 2020.

[24] R. Yamamoto, E. Song, and J. 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)*, May 2020, pp. 6199–6203.

[25] T. Hayashi, "Kan-bayashi/ParallelWaveGAN," 2020. [Online]. Available: https://github.com/kan-bayashi/ParallelWaveGAN

[26] A. Corral, I. Leturia, A. Séguier, M. Barret, B. Dazéas, P. Boula de Mareüil, and N. Quint, "Neural Text-to-Speech Synthesis for an Under-Resourced Language in a Diglossic Environment: The Case of Gascon Occitan," in *Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)*. Marseille, France: European Language Resources Association, 2020, pp. 53–60.

[27] ITU-R, "Recommendation ITU-R BS.1534-3: Method for the subjective assessment of intermediate quality level of audio systems," International Telecommunication Union, Tech. Rep. ITU-R BS.1534-3, 2015.

[28] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426*, 2018. [Online]. Available: https://github.com/lmcinnes/umap

[29] K. Gorman, L. F. E. Ashby, A. Goyzueta, A. D. McCarthy, S. Wu, and D. You, "The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Jul. 2020, pp. 40–50.

[30] S. Moran and D. McCloy, Eds., *PHOIBLE 2.0*. Jena: Max Planck Institute for the Science of Human History, 2019. [Online]. Available: https://phoible.org/