Language-Agnostic Meta-Learning for Low-Resource Text-to-Speech with Articulatory Features

Florian Lux and **Ngoc Thang Vu**
Institute for Natural Language Processing
University of Stuttgart
florian.lux@ims.uni-stuttgart.de

Abstract

While neural text-to-speech systems perform remarkably well in high-resource scenarios, they cannot be applied to the majority of the over 6,000 spoken languages in the world due to a lack of appropriate training data. In this work, we use embeddings derived from articulatory vectors rather than embeddings derived from phoneme identities to learn phoneme representations that hold across languages. In conjunction with language agnostic meta learning, this enables us to fine-tune a high-quality text-to-speech model on just 30 minutes of data in a previously unseen language spoken by a previously unseen speaker.

1 Introduction

The advance of deep learning (Vaswani et al., 2017; Goodfellow et al., 2014) has enabled great improvements in the field of Text-to-Speech (TTS). (Towards-)end-to-end models, such as Tacotron 2 (Wang et al., 2017; Shen et al., 2018), TransformerTTS (Li et al., 2019b), FastSpeech 2 (Ren et al., 2019, 2020), FastPitch (Łańcucki, 2021) and many more famous instances (e.g. Arık et al. (2017) and Prenger et al. (2019)) allow for speech synthesis with unprecedented quality and controllability. The models mentioned here rely on vocoders, such as WaveNet (van den Oord et al., 2016), MelGAN (Kumar et al., 2019), Parallel WaveGAN (Yamamoto et al., 2020) or HiFi-GAN (Kong et al., 2020) to turn the parametric representations that they produce into waveforms. Recently proposed models even include some with the ability to go directly to the waveform from a grapheme or phoneme input sequence, such as EATS (Donahue et al., 2020) or VITS (Kim et al., 2021).

While these methods all perform remarkably well if given enough data, cross-lingual use of data remains a key challenge in TTS. Most modern methods are limited to languages and domains that are rich in resources, which over 6,000 languages are not. Attempts at reducing the required resources in a target language by making use of transfer learning from multilingual data have been made by Azizah et al. (2020); Xu et al. (2020); Chen et al. (2019). The mismatch of input spaces however requires complex architectural changes, which limits their ability to be used in conjunction with other modern TTS architectures. Attempts at fixing the issue of having to transfer knowledge from a source to a target by just jointly training on a mixed set of more and less resource rich languages have been made by He et al. (2021); de Korte et al. (2020); Yang and He (2020), which requires complex training procedures. In this work, we will also attempt to transfer knowledge from a set of high resource languages to a low resource language. We fix previous shortcomings by 1) using a linguistically motivated representation of the inputs to such a system (articulatory and phonological features of phonemes) that enables cross-lingual knowledge sharing and 2) applying the model agnostic meta learning (MAML) framework (Finn et al., 2017) to the field of low-resource TTS for the first time.

Using articulatory features as inputs for neural TTS has been attempted recently by Staib et al. (2020) and Wells et al. (2021), following the classical approach of Jakobson et al. (1961). Both achieved good results when applying this idea to the codeswitching problem, since unseen phonemes in the input space no longer map to nonsensical positions, as it would be the case for the standard embedding-lookup. It has to be noted however, that this only works across languages with similar types of phonemes. Also Gutkin (2017) have applied phonological features to low-resource TTS with fair success. They did however rely on supplementary features, such as dependency parsers and morphological analyzers. Furthermore all of their data and models are proprietary and can therefore not be used to compare results to. In this work, we extend the use of articulatory inputs with

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6858 - 6868

the MAML framework to enable very simple yet well working low-resource TTS that can be applied to almost all modern TTS architectures.

We encounter severe instabilities when using MAML on TTS, which make the standard formulation of MAML infeasible to use. Thus we also propose a modification to MAML, which reduces the procedure's complexity. This allows us to create a set of parameters of a model that can be used to fine-tune to a well working single-language single-speaker TTS model with as little as 30 minutes of paired training data available and even enables zero-shot adaptation to unseen languages. We evaluate the success of our approach with both automatic measures and human evaluation.

Our contributions are as follows: 1) We show that it is beneficial to train a TTS model on articulatory features rather than on phoneme-identities, even in the standard single-language high-resource case; 2) We introduce a training procedure that is closely related to MAML which allows training a set of parameters for a TTS model that can be fine-tuned in a low resource scenario; 3) We provide insights on how much data and training time are required to fine-tune a model across different languages and speakers simultaneously using said meta-parameters; 4) We show that the meta-parameters can generalize to unseen phonemes and rapidly improve their ability to properly pronounce them when fine-tuning. [1]

2 Background and Related Work

2.1 Input Representations

Character Embeddings The simplest approach to representing text as input to a TTS is using indexes of graphemes to look up embeddings. This is however prone to mistakes. Taylor and Richmond (2020) bring up the example of *coathanger*. If the TTS is not aware of the morpheme boundary between the *coat* and the *hang*, it will be inclined to produce something like [kʌθəɪnʤə] rather than the correct [koʊthæŋə]. Such a representation of the input will be highly language dependent, since special pronunciation rules rarely hold for more than a single language.

The textual input can be augmented by adding information, such as morpheme boundaries, intonation phrase boundaries derived from e.g. syntactic parsing as is done in many TTS frontends (Schröder and Trouvain, 2003; Clark et al., 2007; Ebden and Sproat, 2015), or even the semantic identity of the word a character belongs to, using e.g. BERT embeddings (Hayashi et al., 2019).

Phoneme Embeddings Rather than looking up embeddings for graphemes, it is often beneficial to use embeddings of phonemes. Phonemizers (Bisani and Ney, 2008; Taylor, 2005; Rao et al., 2015) produce a sequence of phonetic units, which correlate with the segments in the audio much more than raw text. One such standard of phonetic representation which we make use of is the International Phonetic Alphabet (IPA). Using this set of phonetic units alleviates the problems of TTS fine-tuning and transfer-learning to low-resource domains, because the phonetic units should be mostly language independent. Deri and Knight (2016) provide a data driven approach for the grapheme to phoneme conversion task, which performs well on over 500 languages and can be adapted fairly easily to any new low-resource language. There remains however one major challenge: The use of different phoneme sets for each language, leading to completely unseen units in inference or fine-tuning data.

Latent Representations Li et al. (2019a) claim that multilinguality in speech recognition and TTS can be achieved by changing the input to a latent representation that is trained across languages. While their results seem very promising, their technique needs training data in all languages it should be applied to, which rules out zero-shot settings.

Articulatory Features We fix the shortcoming of not being able to handle unseen phonemes by specifying phonemes in terms of articulatory features such as position (e.g. frontness of the tongue) and category (e.g. voicedness). We show that systems trained on this input can produce a phoneme given nothing but an articulatory description and thus generalize to unseen phonemes. This makes the transfer of knowledge across languages much simpler. A similar approach for the purpose of handling codeswitching has been done in Staib et al. (2020). Our work builds on top of theirs by extending the idea to transfer learning an entire TTS in a new language with minimal data, making use of meta learning on top of articulatory features.

[1] All of our code, as well as the checkpoints for a low-resource fine-tuning capable Tacotron 2 and FastSpeech 2 model are publicly available at https://github.com/DigitalPhonetics/IMS-Toucan.

2.2 Model Agnostic Meta Learning (MAML)

The goal of MAML (Finn et al., 2017) is to find a set of parameters, that work well as initialization point for multiple tasks, including unseen ones. The procedure consists of an outer loop and an inner loop. The outer loop starts with a set of parameters, which we will call the Meta Model. The inner loop trains task specific copies of the Meta Model for a low amount of steps. Once the inner loop is complete, the loss for each of the models is calculated, summed, and backpropagated to the original Meta Model by unrolling the inner loop. This includes the very costly calculation of second order derivatives. The Meta Model is then updated and the inner loop starts again.

This procedure moves the initialization point closer to the optimal configuration for each of the trained tasks, which generalizes to even unseen tasks. Multiple variants of MAML have been suggested that try to fix the high computational cost of the second order derivatives. The simplest one is called first-order MAML and simply applies the gradient of the task specific model at the end of the inner loop directly to the Meta Model. Other variants are described in Antoniou et al. (2019); Rajeswaran et al. (2019).

3 Approach

3.1 System Description

For the implementation of our method, we use the open source IMS Toucan speech synthesis toolkit, first introduced in (Lux et al., 2021), which is in turn based on the ESPnet end-to-end speech processing toolkit (Watanabe et al., 2018; Hayashi et al., 2020, 2021). Neekhara et al. (2021) show, that it is beneficial to fine-tune a single-speaker model to a new speaker rather than to train a multi-speaker model. Inspired by this, we decided to also use a model that is not conditioned on speakers or on languages rather than a conditioned multi-speaker multi-lingual model and fine-tune it on the data from a new speaker in a new language. In preliminary experimentation we got similar results to them within one language, but found their method to not work across languages. In comparison to the fine-tuning of a simple single speaker model, we found training and fine-tuning a model conditioned on language embeddings and speaker embeddings much more sensitive to the choice of hyperparameters. Figure 1 shows an overview of our system, underlining how it is not specific to a certain architecture, but could instead be used in conjunction with almost all modern TTS methods.

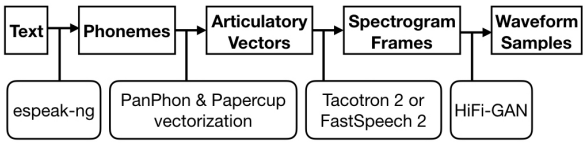


Figure 1: Overview of the TTS pipeline we use. The top row shows the modality in which the data is at this point in the pipeline. The lower row shows the methods that handle the transitions. Each of the blocks in the lower row can be exchanged easily with other methods that have the same interfaces.

Tacotron 2 For our implementation of Tacotron 2 (Shen et al., 2018), we make use of the forward attention with transition agent introduced in Zhang et al. (2018), which uses a CTC-like forward variable (Graves et al., 2006) to promote the quick learning of monotonic alignment between text and speech. To further help with this, we make use of the guided attention loss introduced in Tachibana et al. (2018).

FastSpeech 2 To train the parallel FastSpeech 2 model (Ren et al., 2020), annotations of durations for each phoneme are needed. These also have to be generated for the low-resource fine-tuning data. To that end, we generate alignments using the encoder-decoder attention map of a Tacotron 2 model. Following Kim et al. (2020); Shih et al. (2021); Badlani et al. (2021), we apply the Viterbi algorithm to find the most probable monotonic path through the attention map, which significantly improves the quality of the alignments.

This is especially important, because we train our FastSpeech 2 model with pitch and energy labels that are averaged over the duration of each individual phoneme to allow for great controllability during inference, as is introduced by Łańcucki (2021). Incorrect alignments would lead to follow-up errors such as an unnaturally flat prosody.

Furthermore, we make use of the conformer block (Gulati et al., 2020) as the encoder and decoder, rather than the standard transformer (Vaswani et al., 2017).

3.2 Articulatory Vectors

PanPhon The PanPhon resource (Mortensen et al., 2016) can be used to get linguistic specifications of phonemes. It comes with an open-source

tool[2] which we use to convert phonemes into numeric vectors. Each vector encodes one feature per dimension and takes the value of either -1, 0 or 1, putting the features on a scale wherever meaningful. This featureset also includes phonological features which go beyond simple phonetics, such as whether a phoneme is syllabic.

Papercup Additionally we make use of the purely articulatory description system of phonemes introduced in Staib et al. (2020), which we will call Papercup features in the following. For the encoding we use one-hot vectors, similar to their implementation. Some of the features, like openness or frontness, should be on a scale rather than one-hot encoded. However since the articulatory vector is fed into a fully connected layer, we leave the reconstruction of this dependency between features for the network to learn.

3.3 Language Agnostic Meta Learning

We find that the standard implementation of MAML does not work well for the TTS task. The inner loop needs hundreds of updates in order to make a significant change to the performance of the task specific model. This is probably due to the TTS task being a one-to-many mapping task, where the loss function of measuring the distance to a spectrogram is not an accurate objective for the TTS. For every text, there are infinitely many spectrograms, which could be considered gold data. Those spectrograms could differ in e.g. the speaker who reads the text and how they read the text. Since there are no conditioning signals, the TTS has to update its parameters towards a certain speaker's characteristics in general. However because in our case each task is a different language and a different speaker, the training becomes highly unstable. So ideally we would either need to run MAML's inner loop until convergence, which is generally infeasible, or stabilize the procedure by not allowing the model to adapt further to one task than to the others.

To fix this issue, we calculate the Meta Model's loss on one batch per language. We then sum up the losses, backpropagate and update the Meta Model directly using Adam (Kingma and Ba, 2015). This stabilizes the learning procedure, but still allows the model to update its parameters towards a more universal configuration. Since we have to make this simplification to MAML in order to deal with the different languages as tasks, we call this procedure language agnostic meta learning (LAML). Ultimately, the model should not care about the language it is fine-tuned in, since it should be close to a universal representation of an acoustic model. To give an exact notion of our modifications: We simplified equation 1 to equation 2, where opt is a gradient descent update, B_i is a batch sampled from task i, $\mathcal{L}$ is an objective function, Θ is the set of parameters from the Meta Model and θ_i is the set of parameters specific to task i. To the best of our knowledge, we are the first to successfully apply MAML to TTS with languages being the tasks.

for t steps do:

$$\Theta_t = opt\left(\Theta_{t-1}, \nabla \sum_i \mathcal{L}\left(\theta_{i,d}, B_i\right)\right) \quad (1)$$

where $\theta_{i,d=0} = \Theta_{t-1}$ and for d steps do:

$$\theta_{i,d} = opt\left(\theta_{i,d-1}, \nabla \mathcal{L}\left(\theta_{i,d-1}, B_i\right)\right)$$

for t steps do:

$$\Theta_t = opt\left(\Theta_{t-1}, \nabla \sum_i \mathcal{L}\left(\Theta_{t-1}, B_i\right)\right) \quad (2)$$

4 Experiments

In this section we will go over the experiments we conducted. First we will evaluate the articulatory features on their own in a single language setting using automatic measures. Then we will evaluate the combination of LAML and articulatory features in a cross-lingual setting using both automatic measures and human evaluation.

In our experiments we make use of the following datasets: The English Nancy Krebs dataset (16h) from the Blizzard challenge 2011 (Wilhelms-Tricarico et al., 2011; King and Karaiskos, 2011); The German dataset of the speaker Karlsson (29h) from the HUI-Audio-Corpus-German (Puchtler et al., 2021); The Greek (4h), Spanish (24h), Finnish (11h), Russian (21h), Hungarian (10h), Dutch (14h) and French (19h) subsets of the CSS10 dataset (Park and Mulc, 2019).

4.1 Mono-Lingual Experiments

4.1.1 Embedding Function Design

To explore our first hypothesis, we investigate the capabilities of the articulatory phoneme representations to be used in a single-speaker and single-language TTS system. To compare different ways

[2] https://github.com/dmort27/panphon

of embedding the features, we train only the embedding function. As gold data we use the embeddings from a well trained lookup-table based Tacotron 2 model. In table 1 we show the average distances of all articulatory vectors as projected by the embedding function to their identity based embedding counterpart. The distance d between two embedding vectors A and B is defined in equation 3.

$$d = \left(\sum_i |A_i - B_i| \right) - \frac{\sum_i A_i \cdot B_i}{\sqrt{\sum_i A_i^2} \cdot \sqrt{\sum_i B_i^2}} \quad (3)$$

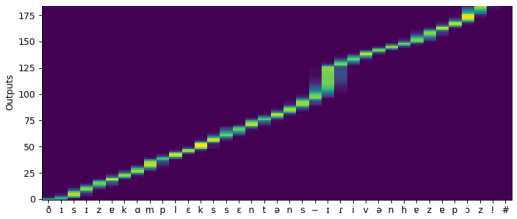
This distance function is also used as the objective function. The embedding functions are each trained for 3000 epochs using Adam (Kingma and Ba, 2015) with a batchsize of 32. The first column shows the results of the articulatory features being fed into a linear layer that projects them into a 512 dimensional space. The second column shows the results of the articulatory features being fed into a linear layer that projects them into a 100 dimensional space, applies the tanh activation function and then further projects them into a 512 dimensional space. As can be seen from the results, it is beneficial to both concatenate the PanPhon features with the Papercup features despite their overlap and to add a nonlinearity into the embedding function to match the embeddingspace of a well trained Tacotron 2 model. Hence we use this setup in all following experiments.

d	Linear	Non-Linear
PanPhon	0.47	0.1
Papercup	0.44	0.05
Combined	0.4	**0.001**

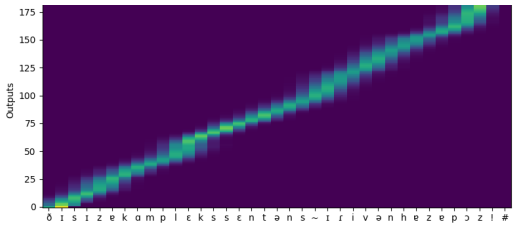
Table 1: Average distance of all embedded articulatory vectors to their position in an embedding space learned in a lookup-table based model.

4.1.2 Convergence Time

To investigate the impact that the articulatory features have on their own, we train a Tacotron 2 with and without them on the Nancy dataset and compare their training time and final quality. While the model trained on embedding tables shows a clear diagonal alignment of text and spectrogram frames on an unseen test sentence after 2,000 steps, the one trained on articulatory features does so already at 500 steps. This is visualized in figure 2. The decoder of the Tacotron 2 model can only start to learn to decode after the alignment of inputs to outputs is learned. So learning the alignment earlier gives the articulatory model a clear benefit. After training for 80,000 steps however, our own subjective assessment finds no difference in quality between the two. The earlier convergence of the alignment however shows a possible advantage of using the articulatory features on low-resource tasks, as quicker training progress means that training can be stopped earlier, before overfitting on little data becomes too problematic.



(a) Proposed Tacotron 2 with articulatory features at 500 steps with a batchsize of 32.



(b) Baseline Tacotron 2 with embedding-lookup at 2000 steps with a batchsize of 32.

Figure 2: The first instance of diagonal encoder-decoder attention on an unseen test sentence.

4.2 Cross-Lingual Experiments

In order to investigate the effectiveness of our proposed LAML procedure, we train a Tacotron 2 model and a FastSpeech 2 model on the full Karlsson dataset as a strong baseline. We also train another Tacotron 2 model and another FastSpeech 2 model on speech in 8 languages with one speaker per language (Nancy dataset and CSS10 dataset) and fine-tune those models on a randomly chosen 30 minute subset from the Karlsson dataset. To our surprise, we did not only match, but even outperform the model trained on 29 hours with the model fine-tuned on just 30 minutes in multiple metrics.

As a second baseline we tried to train another meta-checkpoint using the embedding lookup-table approach to also further investigate the effectiveness of the articulatory features. We did however not manage to get such a model to converge to a usable state. This already shows the superiority of

the articulatory feature representations for such a multilingual use-case.

Furthermore we tried to fine-tune the well trained English single speaker models from the first experiment on the 30 minutes of German to have another baseline that can be used to measure the impact of the LAML procedure. This setup however also did not yield any usable results. During the fine-tuning process, the model was capable of speaking German with a strong English accent, yet it did not properly learn to speak in the voice of the target speaker. By the time the model learned to speak in the new speaker's voice, it had overfitted the 30 minutes of training data and collapsed, producing no more intelligible speech. We conclude that the method proposed in this paper not only improves on the ability to use cross-lingual data easily, but actually enables it in the first place. Both the articulatory features, as well as the LAML pretraining seem necessary to achieve cross-lingual fine-tuning on low-resource data.

The texts we use for the following experiments are disjunct from any training data used. Human speech as gold standard is not used, since we are interested in the difference in performance between the systems, not their absolute performance. The close to state-of-the-art performance of the baselines is considered as given, considering their ideal training conditions and use of proven methods. Furthermore, we chose to use German as our benchmark language over an actual low-resource language, since it is much easier to acquire reliable ratings on intelligibility and naturalness for German, than it would be for an actual low-resource language.

4.2.1 Intelligibility

To compare intelligibility between our baseline models and our low-resource models, we use the word error rate (WER) of an automatic speech recognition system (ASR) as a proxy. We synthesize 100 sentences of German radio news texts taken from the DIRNDL corpus (Eckart et al., 2012) with each of our baselines and corresponding low-resource systems. Table 2 shows WERs that the German IMS-Speech ASR (Denisov and Vu, 2019) achieves on the synthesized data. For both Tacotron 2 and the FastSpeech 2 based system, the WER of the low-resource model is slightly lower than that of the baseline, thus the low-resource models performed slightly better.

Looking into the cases where the low-resource system outperformed the baseline, we find code-switched segments, where the texts contain names of Russian cities. Since the pretraining data of the low-resource model includes Russian speech, it seems to have not forgotten entirely about what it has seen in the pretraining phase, which in our interpretation confirms the effectiveness of the LAML against the catastrophic-forgetting problem (French, 1999) of regular pretraining.

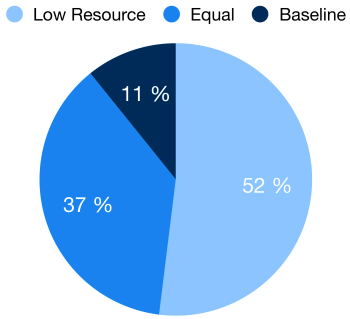
WER	Baseline	Low-Resource
Tacotron 2	13.1%	12.7%
FastSpeech 2	9.9%	9.7%

Table 2: WER of the synthesis systems on 100 radio news texts measured using the IMS-Speech ASR.

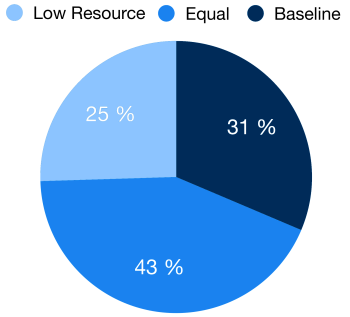
4.2.2 Naturalness

In order to assess the naturalness of the fine-tuned models, we conduct a preference study with 34 native speakers of German. Each participant is shown 12 phonetically balanced samples produced by the Tacotron 2 and FastSpeech 2 models. For every sentence, there is one sample produced by the baseline and one by the low-resource model. The participants are then asked to indicate their subjective overall preference between the two samples. The results for Tacotron 2 are shown in figure 3 (a). The low-resource system was the preferred system in more than half of the cases, with an equal rating taking up more than another third, showing a clear preference for the low-resource model over the baseline. The results for FastSpeech 2, as seen in figure 3 (b), are a lot more balanced. While the baseline is preferred more often than the low-resource variant, it is not the case in the majority of the ratings. In 56% of the cases, the model fine-tuned on 30 minutes of data was perceived to be as good or better than the model trained on 29 hours.

Computational Resources All models were trained on a single NVIDIA A6000 GPU. Training the Tacotron Baseline took 2 days. Training time of the FastSpeech Baseline was 1 day. Training time of the meta-checkpoint was 4 days, finetuning to a new model from the meta-checkpoint however only takes 2 hours. The HiFi-GAN vocoder used to generate all samples took 4 days to train and was not fine-tuned on the unseen data. We did not perform hyperparameter searches and used the suggested default settings for all methods, which worked sufficiently well, but could surely be improved.



(a) Preference ratings for 102 Tacotron 2 samples.



(b) Preference ratings for 102 FastSpeech 2 samples.

Figure 3: Results of the preference study comparing a low-resource model to a high-resource baseline.

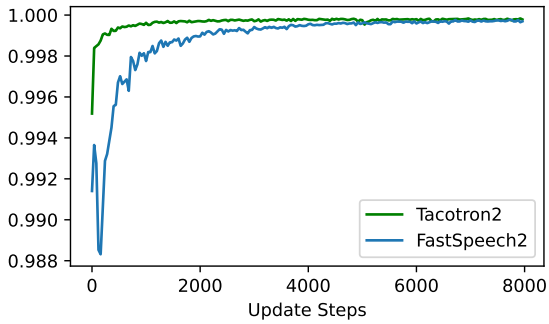


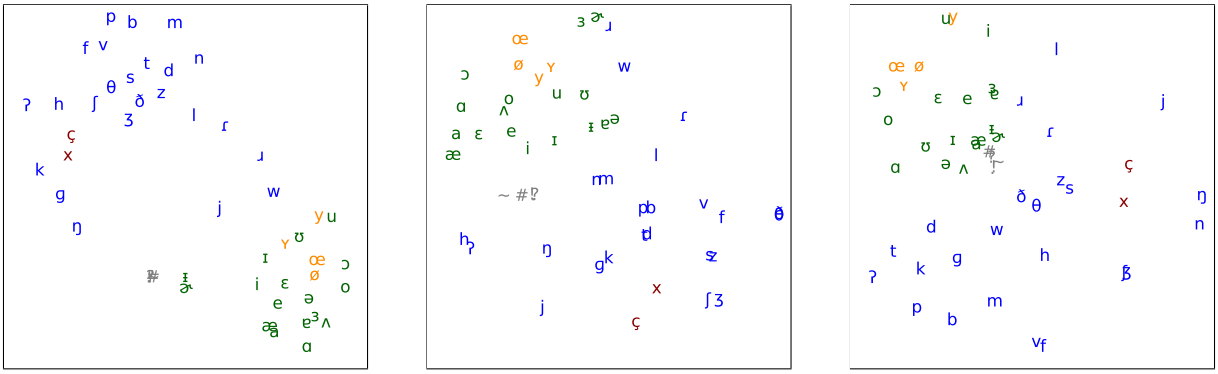
Figure 4: Cosine similarity of speaker embeddings to target speaker over time.

5 Further Analysis and Future Work

What is the ideal amount of training steps for fine-tuning? To investigate the amount of update steps needed to fully adapt to the new speaker with the added difficulty of learning a new language, we show the cosine similarity of a speaker embedding of the fine-tuned model to that of the ground truth throughout the fine-tuning process in figure 4. The speaker embedding is built according to the ECAPA-TDNN architecture (Desplanques et al., 2020) and provided open source by SpeechBrain (Ravanelli et al., 2021). It is trained on VoxCeleb 1 and 2 (Nagrani et al., 2017, 2019; Chung et al., 2018) which to the best of our knowledge does not overlap with any of the other training and evaluation data we used. We tried to decrease adaptation time further by incorporating said speaker embedding similarity as an additional objective function, similar to Nachmani et al. (2018), we did however see only marginal improvements in the amount of steps needed at the expense of greatly increased training time.

Can this setup handle zero-shot phonemes? We show the model's zero shot capabilities in figure 5. We removed Dutch and Finnish from the training data of the meta-checkpoint and trained another version of it, to be able to see how it handles all of the now completely unseen phonemes specific to German. While their correct position in plot (a) can be considered given, since it shows the articulatory featurespace, their meaningful positions in plot (b) and (c) show that the meta-checkpoint does not just collapse the vector of the unseen phoneme to the one it is most similar to, but actually generalizes. While their pronunciation when produced does not match the correct pronunciation perfectly, it can be understood in the context of a longer sequence. This is congruent with the results of Staib et al. (2020). During the adaptation phase, the pronunciation of the unseen phonemes rapidly matches the correct pronunciation after less than 100 steps.

Does this setup learn the difference between language and speaker? When analyzing the fine-tuned meta-checkpoint, we observed that it seems to link the language of the input to the voice of the speaker. For example when synthesizing an unseen Hungarian text using Tacotron 2, the voice of the synthesis resembles that of the Hungarian female speaker, even though the model has been fine-tuned on the male German speaker and there are no additional conditioning signals. We hypothesize that the LAML procedure induces certain subsets of parameters in the model to be speaker dependent and the encoder of the model priming those parameters purely based on the phoneme sequence. This leads us to believe, that the fine-tuning of all parameters in the model may neither be necessary, nor even the best way of adapting to new data. This also fits the observations of the speaker embedding over time, since the Tacotron model adapts to the new speaker very rapidly. Further investigations into



(a) 66 dimensional Featurespace (24 PanPhon and 42 Papercup features)

(b) 512 dimensional embeddingspace learned during Tacotron 2 training

(c) 384 dimensional embeddingspace learned during FastSpeech 2 training

Figure 5: t-SNE visualizations of phoneme representations, illustrating zero-shot capabilities. Special characters are grey, consonants are blue, vowels are green, unseen consonants are red and unseen vowels are orange.

the interactions between parameter groups could allow cutting down the amount of parameters that need to be trained significantly, further reducing the need for training data.

How can we bring down FastSpeech 2's data need further? A similar observation regarding language and speaker can be made with FastSpeech 2, however as could be seen from the experiment on naturalness and the training time, the FastSpeech 2 model can benefit more from additional data and training time. This may come down to its nearly twice as high parameter count. So a more effective fine-tuning strategy, that considers some parameters as constants, could benefit the fine-tuning capabilities of the FastSpeech 2 model greatly.

Does this work across language families? One limitation to our findings is that we investigated only the transfer of languages that share similar phoneme inventories. It is possible that fine-tuning to a language that uses e.g. the lexical tone rather than pitch accents or word accents would require pretraining in more closely related high-resource languages, such as Chinese. However, as Vu and Schultz (2013) find in their analysis of multilingual ASR, the fast adaptation of an acoustic model trained on multiple languages to unseen languages works well, even across different language families. We thus believe that the technique and analysis presented in this paper also holds across language families and types.

6 Conclusion

In this paper, we show an approach for training a model in a language for which only 30 minutes of data are available by making use of articulatory features and language agnostic meta learning. The main takeaways from our work are as follows:

Articulatory Features for TTS Using articulatory features as the input representation to a TTS system enables the use of multilingual data without the need for increased architectural complexity, such as language specific projection spaces. It is furthermore beneficial to use even in single-language scenarios, since the knowledge sharing between phonemes makes the TTS system converge much earlier to an usable state during training.

MAML on TTS Applying MAML to TTS does not work well. If we however remove the inner loop, we are able to pretrain a low-resource capable checkpoint for TTS. This modification not only makes it work, it also simplifies the formulation.

Zero-shot capabilities The use of articulatory features enables zero-shot inference on unseen phonemes. This is further enhanced by the LAML training procedure. The implications of this are particularly interesting for codeswitching, as Staib et al. (2020); Wells et al. (2021) have pointed out previously. Using these two techniques in conjunction could be used to reduce the problem of codeswitching to a problem of token-wise language identification.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful feedback and suggestions. This work was funded by the Carl Zeiss Foundation.

References

Antreas Antoniou, Harri Edwards, and Amos Storkey. 2019. How to train your MAML. In *Seventh International Conference on Learning Representations*.

Sercan O. Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. 2017. Deep Voice: Real-time Neural Text-to-Speech. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, volume 70 of *Proceedings of Machine Learning Research*. PMLR.

Kurniawati Azizah, Mirna Adriani, and Wisnu Jatmiko. 2020. Hierarchical Transfer Learning for Multilingual, Multi-Speaker, and Style Transfer DNN-Based TTS on Low-Resource Languages. *IEEE Access*, 8:179798–179812.

Rohan Badlani, Adrian Łancucki, Kevin J Shih, Rafael Valle, Wei Ping, and Bryan Catanzaro. 2021. One TTS alignment to rule them all. *arXiv preprint arXiv:2108.10447*.

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

Yuan-Jui Chen, Tao Tu, Cheng-chieh Yeh, and Hung-Yi Lee. 2019. End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning. *Proc. Interspeech 2019*, pages 2075–2079.

J. S. Chung, A. Nagrani, and A. Zisserman. 2018. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*.

Robert AJ Clark, Korin Richmond, and Simon King. 2007. Multisyn: Open-domain unit selection for the Festival speech synthesis system. *Speech Communication*, 49(4):317–330.

Marcel de Korte, Jaebok Kim, and Esther Klabbers. 2020. Efficient Neural Speech Synthesis for Low-Resource Languages Through Multilingual Modeling. *Proc. Interspeech 2020*, pages 2967–2971.

Pavel Denisov and Ngoc Thang Vu. 2019. IMS-speech: A speech to text tool. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pages 170–177.

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

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pages 3830–3834. ISCA.

Jeff Donahue, Sander Dieleman, Mikolaj Binkowski, Erich Elsen, and Karen Simonyan. 2020. End-to-end Adversarial Text-to-Speech. In *International Conference on Learning Representations*.

Peter Ebden and Richard Sproat. 2015. The Kestrel TTS text normalization system. *Natural Language Engineering*, 21(3):333–353.

Kerstin Eckart, Arndt Riester, and Katrin Schweitzer. 2012. A discourse information radio news database for linguistic analysis. In *Linked Data in Linguistics*, pages 65–76. Springer.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

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

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. *Proc. Interspeech 2020*, pages 5036–5040.

Alexander Gutkin. 2017. Uniform Multilingual Multi-Speaker Acoustic Model for Statistical Parametric Speech Synthesis of Low-Resourced Languages. *Proc. Interspeech 2017*, pages 2183–2187.

Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Shubham Toshniwal, and Karen Livescu. 2019. Pre-Trained Text Embeddings for Enhanced Text-to-Speech Synthesis. In *INTERSPEECH*, pages 4430–4434.

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. 2020. ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7654–7658. IEEE.

Tomoki Hayashi, Ryuichi Yamamoto, Takenori Yoshimura, Peter Wu, Jiatong Shi, Takaaki Saeki, Yooncheol Ju, Yusuke Yasuda, Shinnosuke

Takamichi, and Shinji Watanabe. 2021. ESPnet2-TTS: Extending the Edge of TTS Research. *arXiv preprint arXiv:2110.07840*.

Mutian He, Jingzhou Yang, and Lei He. 2021. Multilingual Byte2Speech Text-To-Speech Models Are Few-shot Spoken Language Learners. *arXiv preprint arXiv:2103.03541*.

Román Jakobson, C. Gunnar M. Fant, and Morris Halle. 1961. Preliminaries to Speech Analysis: The Distinctive Features and Their Correlates.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Simon King and Vasilis Karaiskos. 2011. The Blizzard Challenge 2011. In *Proc. Blizzard Challenge Workshop*, volume 2011.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. *Advances in Neural Information Processing Systems*, 32.

Adrian Łańcucki. 2021. FastPitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan. 2019a. Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5621–5625. IEEE.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019b. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713.

Florian Lux, Julia Koch, Antje Schweitzer, and Ngoc Thang Vu. 2021. The IMS Toucan system for the Blizzard Challenge 2021. In *Proc. Blizzard Challenge Workshop*, volume 2021. Speech Synthesis SIG.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. PanPhon: A Resource for Mapping IPA Segments to Articulatory Feature Vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf. 2018. Fitting new speakers based on a short untranscribed sample. In *International Conference on Machine Learning*, pages 3683–3691. PMLR.

A. Nagrani, J. S. Chung, and A. Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2019. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*.

Paarth Neekhara, Jason Li, and Boris Ginsburg. 2021. Adapting TTS models For New Speakers using Transfer Learning. *arXiv preprint arXiv:2110.05798*.

Kyubyong Park and Thomas Mulc. 2019. CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages. *Proc. Interspeech 2019*, pages 1566–1570.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. WaveGlow: A flow-based generative network for speech synthesis. In *ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Pascal Puchtler, Johannes Wirth, and René Peinl. 2021. Hui-audio-corpus-german: A high quality tts dataset. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 204–216. Springer.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 113–124.

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

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh,

Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. 2021. SpeechBrain: A general-purpose speech toolkit. ArXiv:2106.04624.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. In *International Conference on Learning Representations*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. FastSpeech: fast, robust and controllable text to speech. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3171–3180.

Marc Schröder and Jürgen Trouvain. 2003. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS Synthesis by Conditioning WaveNet on mel Spectrogram Predictions.

Kevin J Shih, Rafael Valle, Rohan Badlani, Adrian Lancucki, Wei Ping, and Bryan Catanzaro. 2021. RAD-TTS: Parallel Flow-Based TTS with Robust Alignment Learning and Diverse Synthesis. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*.

Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, Devang S. Ram Mohan, Lorenzo Foglianti, Raphael Lenain, and Jiameng Gao. 2020. Phonological Features for 0-Shot Multilingual Speech Synthesis. *Interspeech 2020*.

Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. 2018. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4784–4788. IEEE.

Jason Taylor and Korin Richmond. 2020. Enhancing Sequence-to-Sequence Text-to-Speech with Morphology. In *INTERSPEECH*, pages 1738–1742.

Paul Taylor. 2005. Hidden Markov models for grapheme to phoneme conversion. In *Ninth European Conference on Speech Communication and Technology*. Citeseer.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ngoc Thang Vu and Tanja Schultz. 2013. Multilingual multilayer perceptron for rapid language adaptation between and across language families. In *Interspeech*, pages 515–519.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards End-to-End Speech Synthesis. *Proc. Interspeech 2017*, pages 4006–4010.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Dan Wells, Pilar Oplustil-Gallegos, and Simon King. 2021. The CSTR entry to the Blizzard Challenge 2021. In *Proc. Blizzard Challenge Workshop*, volume 2021. Speech Synthesis SIG.

Reiner Wilhelms-Tricarico, Brian Mottershead, Rattima Nitisaroj, Michael Baumgartner, John Reichenbach, and Gary Marple. 2011. The lessac technologies system for blizzard challenge 2011. In *Blizzard Challenge 2011 Workshop paper. DOI= http://festvox.org/blizzard/bc2011/LESSAC Blizzard2011. pdf*. Citeseer.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. *LRSpeech: Extremely Low-Resource Speech Synthesis and Recognition*, page 2802–2812. Association for Computing Machinery, New York, NY, USA.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. 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)*, pages 6199–6203. IEEE.

Jingzhou Yang and Lei He. 2020. Towards Universal Text-to-Speech. In *INTERSPEECH*, pages 3171–3175.

Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai. 2018. Forward attention in sequence-to-sequence acoustic modeling for speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4789–4793. IEEE.