Multilingual Speech Synthesis Using DNN Model for French and English Languages

Rougié Tanguy

岡山大学 阿部研究室

Abe Laboratory, Okayama University

Summary

In this study, we aim to develop a multilingual speech synthesis model using a deep neural network. We train the network on both English and French to synthesize English with an English accent and French with a French accent, as well as French with an English accent and vice versa. The VITS model will be extensively used as it provides the best synthesis results. The primary goal is to create a multilingual model that maximizes speaker similarity. A potential future application could be synthesizing speech in one language from text in another language, then resynthesizing it with the appropriate accent in the original language.

1 Introduction

Speech synthesis has been extensively developed, particularly with services like Alexa, Google Home, Siri, and text-to-speech functionalities. However, there are still many possible improvements in terms of voice quality and naturalness (making it more natural and intonations), expressiveness and personalization (controlling prosody, personalized speech synthesis), contextual synthesis and implementing emotions (modify the intonation with the context), multilingual synthesis, and accent recognition/synthesis, which is the focus of this study.

The main challenge is to design a single model that includes language and speaker information. To achieve this, we rely on the VITS model, ensuring initially that it supports speech synthesis in both target languages. Then, we modify it to incorporate language management.

2 Database Acquisition

For the dataset, an initial idea was to use a speech dataset generator, which can convert YouTube videos into .wav audio files with a chosen sampling rate. By selecting YouTube videos with familiar voices, it becomes easier to recognize similarities and perceive differences, which is more challenging with unfamiliar voices. To achieve this, we retrieve videos and partition them into audio files of about ten seconds each.

However, there are three main difficulties: first, for precise synthesis, a training set of several dozen hours with varied vocabulary is needed, which is hard to obtain from YouTube videos. Second, the videos must contain only the target voice, and third, the audio must be accurately converted into text. For these reasons, the M-AILABS dataset was used instead. For this study, we utilized 29 hours of audio from a single French speaker and 32 hours of audio from a single North American speaker.

The dataset contains some inconsistencies, particularly with certain apostrophes disappearing (for example, « C'était » becomes « Cétait »). Therefore, the first column containing the raw, unprocessed sentences was initially chosen. However, the problem is that quotation marks can appear, causing issues during the creation of the lists. Ultimately, the decision was made to use the second column, which removes the problematic characters.

3 French Speaker model

3.1 Filelists creation

To train the model, the dataset needs to be divided into three parts: the training set (95% of the dataset), the validation set (<1% of the dataset), which is used to adjust the parameters at each epoch, and the test set, which is used at the end of the training to estimate the model's efficiency. The first step is to randomly shuffle the datasets and distribute them into these three files. Then, preprocessing must be done, which means transforming the text into the target language's phonemes, as the model takes phonemes as input. To do this, the symbol definition file must be adapted to include all phonemes specific to the language (for French, this simply involves adding accented letters compared to English).

During this step, warnings are raised due to the mismatch between the number of words before and after the transformation. This is because phonetic transcription does not always adhere to word segmentation. Additionally, there is a significant variability in the time taken for next transformation: with each execution of the preprocessing, the first sentences are preprocessed much more quickly, and the preprocesing time seems to increase quadratically with the number of sentences processed. Commands to see which function the program is in it at each moment have not yet helped identify the cause of this phenomenon.

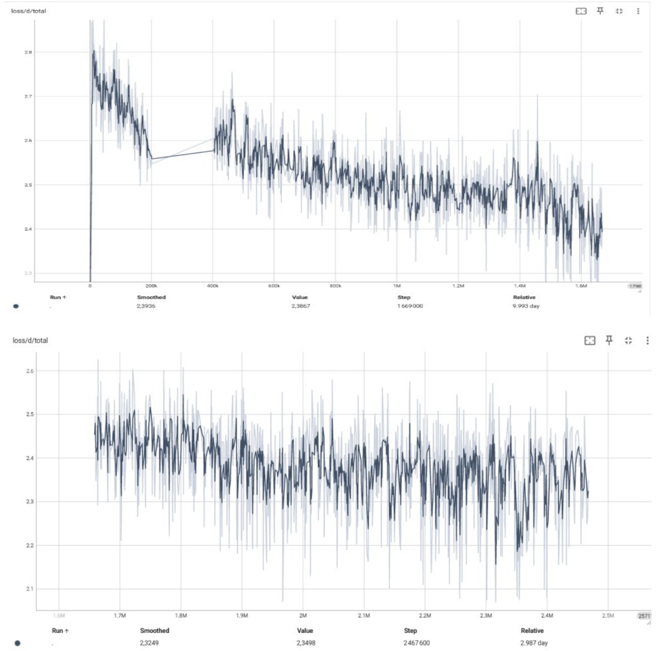
3.2 Model adjustements

Several modifications were made to the models and hyperparameters. The sampling rate was set to 16kHz, the evaluation interval was extended to avoid file overload, and the batch size was reduced to 32 due to a lack of memory for higher values. The text_cleaners used in preprocessing and for phoneme transformation utilizes the phonemize function from the phonemizer library, with the next being converted to ASCII beforehand.

After training for a thousand epochs, testing sentences to evaluate the model's effectiveness revealed that accented letters are pronounced as they had no accent. The issue arises because the conversion of the text to ASCII character is coded in 7 bits, which means only the first 127 characters of the table are used. However, accented letters, such as 'é', are coded in 8 bits. Therefore, the conversion function in the text cleaner needs to be removed.

3.3 Results and Analysis

We can plot the loss function using TensorBoard, resulting in the data shown in Figure 1. It is observed that before processing, the cost function reaches 2.39, and after processing, it reaches 2.32. This suggests that handling accents is likely the cause of these differences. The jump from 200k steps to 400k steps is due to doubling the dataset at that point without changing any other hyper parameters, effectively doubling the number of steps. We can also plot the mel-spectrogram, where it is noticeable that stripes appear after accent processing. Therefore, the improvement of the model is evident.



Run ↑	Smoothed	Value	Step	Relative
.	2,3936	2,3867	1 669 000	9.993 day

Run ↑	Smoothed	Value	Step	Relative
.	2,3249	2,3498	2 467 600	2.987 day

Figure 1: Loss function over epochs with incorrect processing (top) and with correct processing (bottom)

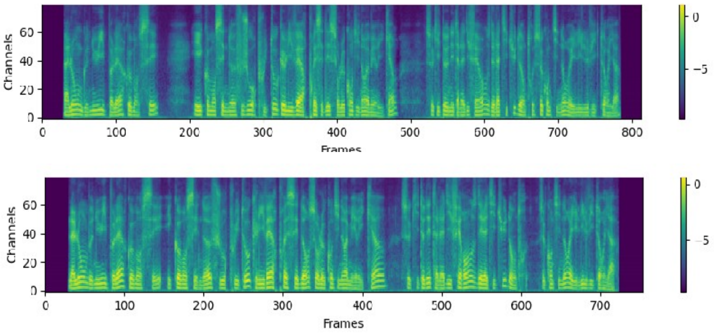
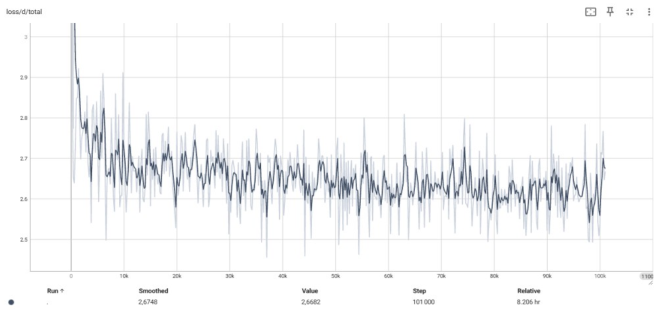


Figure 2: Mel-spectrogram with incorrext processing (top) and with correct processing (bottom)

4 Multilingual model

The first approach to handling the multilingual model is to consider the multi-speaker model by selecting a French speaker as voice one and English speaker as voice two. The problem with this method is that it trains on the speakers rather than on the languages, which may lead to poor results. However, this will be our starting point to compare the cost function values. The steps are quite similar to the single-speaker approach, with the difference being that a speaker ID is added to the second column of the data. It is also necessary to ensure that the sampling rate is the same. The two datasets are transformed separately according to their text_cleaner, then they are mixed to form the three files: train, test, and val.

By visualizing the model's characteristics with TensorBoard, we notice that the cost function is higher than for monolingual model and decreases very slowly. Additionally, the mel-spectrogram is very atypical and does not seem to reflect good behavior. When testing the model with French and English sentences, we observe that English is synthesized very well, whereas French is much less so. However, upon closer inspection, the difference is not at the language level but at the speaker level. Synthesizing French with the English speaker results in an understandable sentence, though pronounced with an accent. This issue might be due to the text_cleaner in the configuration file, which is used for English but should not be applied if we do not want to process the text at this level. The next step is to use model that implements a language-related module



Run ↑	Smoothed	Value	Step	Relative
.	2,6748	2,6682	101 000	8.206 hr

Figure 3: Loss fuction over epochs for multilingual model

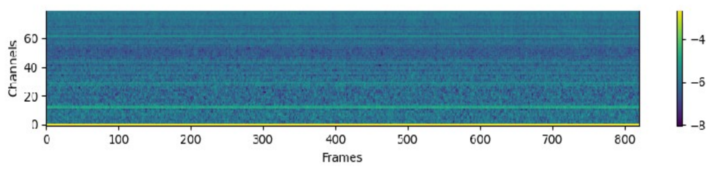


Figure 4: Mel-spectrogram for multilingual model

5 Conclusion and next steps

The results obtained are very satisfactory for the monolingual model but not for the multilingual model. However, the model used is still capable of synthesizing accents, which could be interesting to study further, especially if data on different regional accents for the same language is available. Before exploring this, the next phase of the study will focus on an improved and more complex model.

6 References

[1] Jaehyeon Kim, Jungil Kong, Juhee Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech" Jun 11, 2021

[2] E. Casanova, J. Weber, C. Shulby, A. C. Junior, E. Gölge, M. A. Ponti, "YourTTS: Towards Zero-shot Mulispeaker TTS and zero-Shot Voice Conversion for Everyone" Dec 13, 2023

[3] Munich Artificial Intelligence Laborattories GmbH. The m-ailabs speech dataset - caito, 2017. https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/.

[4] David Martin Rius, "Speech Dataset Generator", Feb 2024, https://github.com/davidmartinrius/speech-dataset-generator

[5] M. Hansen, "Piper" 2021 https://github.com/rhasspy/piper

[6] Srinivas Billa, "efficient-vits-finetuning" 2021 https://github.com/nivibilla/efficient-vits-finetuning