eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Basque and Spanish Multilingual TTS Model for Speech-to-Speech Translation

Author: Xabier de Zuazo
Advisors: Dr. Ibon Saratxaga, PhD

Dr. Eva Navas, PhD

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

February 2023

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Acknowledgements

The research conducted here would not have been possible to complete without the clear guidance and constant recommendations of the tutors: Ibon Saratxaga and Eva Navas. They provided me with all the resources required and helped me constantly along the way. Also, I would like to thank Inma Hernáez for her recommendations and testing of the platform and models.

Similarly, I cannot leave without thanking the evaluators who participated in the tests and the Aholab team, who have been the cornerstone in obtaining reliable results and being able to conclude the work satisfactorily. At this point, I cannot forget to mention Marko Txopitea either, for his help in testing some of the transcription models, and for providing very detailed and revealing feedback.

Besides, this research has been carried out on the shoulders of giants, some of whom I was lucky enough to lend me a hand. Among the community of researchers embracing free software and transparency, I would like to give a special thanks to Francis M. Tyers and Edresson Casanova for their hard research work, answering my doubts and showing me the path to follow.

Last but not least, I am extending my sincere gratitude to my family and close friends for their patience and understanding during this period of hard work.

Laburpena

Azkenaldian, Text-to-Speech eredu anitz sortu dira sare neuronal sakonak erabiliz, testutik audioa sintetizatzeko. Lan honetan, state-of-the-art Text-to-Speech eredu eleaniztun eta hiztun anitzeko eredua landu da euskaraz, gaztelaniaz, katalanez eta galegoz. Ikerketa honetan datu-multzoak bildu, haien audio- eta testu-datuak aldez aurretik prozesatu, eredua hizkuntzetan entrenatu da urrats desberdinetan eta emaitzak puntu bakoitzean ebaluatu dira. Entrenatze-urratserako, ikaskuntza-transferentzia teknika erabili da dagoeneko hiru hizkuntzatan trebatutako eredu batetik abiatuta: ingelesa, portugesa eta frantsesa. Beraz, hemen sortutako azken ereduak zazpi hizkuntza onartzen ditu guztira. Gainera, eredu hauek zero-shot ahots bihurketa ere egiten dute, sarrerako audio fitxategi bat erreferentzia gisa erabiliz. Azkenik, Speech-to-Speech Translation egiteko prototipo aplikazio bat sortu da hemen entrenatutako ereduak eta komunitateko beste eredu batzuk elkartuz. Bide horretan, Deep Speech Speech-to-Text eredu batzuk sortu dira euskararako eta galegorako.

Abstract

Lately, multiple Text-to-Speech models have emerged using Deep Neural networks to synthesize audio from text. In this work, the state-of-the-art multilingual and multi-speaker Text-to-Speech model has been trained in Basque, Spanish, Catalan, and Galician. The research consisted of gathering the datasets, pre-processing their audio and text data, training the model in the languages in different steps, and evaluating the results at each point. For the training step, a transfer learning approach has been used from a model already trained in three languages: English, Portuguese, and French. Therefore, the final model created here supports a total of seven languages. Moreover, these models also support zero-shot voice conversion, using an input audio file as a reference. Finally, a prototype application has been created to do Speech-to-Speech Translation, putting together the models trained here and other models from the community. Along the way, some Deep Speech Speech-to-Text models have been generated for Basque and Galician.

Keywords: multilingual multi-speaker Text-To-Speech, Speech-to-Text, Machine Translation, Speech-to-Speech Translation, cross-lingual zero-shot voice conversion, Basque, Spanish.

Contents

List of Figures

List of Tables

1 Introduction

Speech-to-Speech Translation systems (S2ST) are used to translate speech from a specific language to another different language (Arora et al., 2013). The main goal of these systems is to help people who do not speak a common language to communicate or find it easier to express themselves in another language. Another possible purpose is to reduce the amount of data to be transmitted, for example, when the distance is long or the communication is very unstable.

Traditionally, these systems use an architecture that concatenates different modules in a cascade manner (Lavie et al., 1997; Lazzari, 2006). Specifically, it consists of splitting the task into three simpler sub-tasks: speech recognition, text translation, and then its synthesis to produce speech again. In this research, a first prototype of a Speech-to-Speech system is designed using this traditional approach. The design of a modularized structure comes with the added advantage that, in addition to voice translation, the modules can also be used separately for other future tasks. For example, in the case that concerns us, as will be shown below, the developed transcription system will also be used to evaluate the synthesis model and decide the best path to improve it.

Additionally, and in our case, the models can also conserve the speaker's voice from the input to the output waveform by using a voice identifier vector extracted from the audio. In Figure 1, there is a simplified diagram of a Speech-to-Speech Translation model with architecture in cascade. In the next subsections, each of the modules will be introduced.

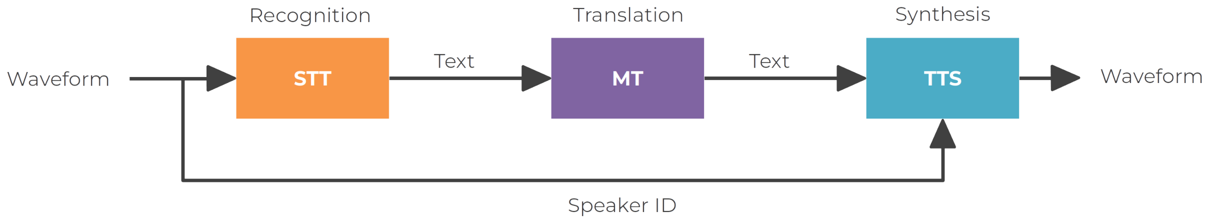


Figure 1: Speech-to-Speech Translation model diagram with voice conversion support.

The following is a brief description of each of the models that is part of the modular architecture of a Speech-to-Speech solution.

1.1 Text-to-Speech (TTS) Models

The study of Text-to-Speech systems (TTS), also known as speech synthesizers, is a vitally important field of speech processing and a fundamental piece of human-computer interaction (Paul, 2014). The main goal of a Text-to-Speech system is to produce human-like speech and mimic real-human speakers starting from a text-formatted input. In other words, it makes computer systems able to read and pronounce texts like humans. The basic task of these systems is to produce sound waves that human listeners easily understand. In Figure 2, there is an example of a waveform generated by synthesizing a sentence.

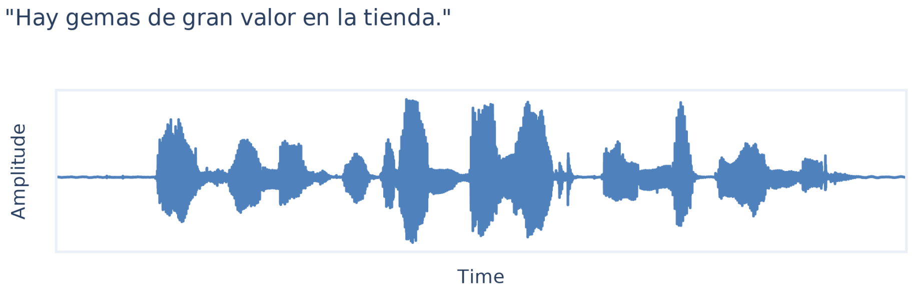


Figure 2: Generated waveform example by synthesizing a sentence.

Today, there are already multiple approaches and working solutions for the task. These systems are beginning to aspire to higher objectives, such as producing indistinguishable sounds from real humans, adding prosodic features, or imitating specific speaker voices. When a TTS system supports multiple languages, it is known as a multi-language TTS model. If it supports the synthesis using different speakers, it is called a multi-speaker model. Additionally, some may be able to produce speech from a speaker not known in advance by the model: this is called a zero-shot multi-speaker TTS model.

Speech synthesis has multiple uses, for example, applications that speak to people, like conversation agents, navigation systems, or helping blind people to browse the internet and play video games. Another different use is to help people speak, for example, sufferers of neurological disorders or individuals who have undergone laryngectomies. In general, helping humans who have lost the ability to speak for any reason but are fully capable of typing text through some interface.

1.2 Speech-to-Text (STT) Models

Speech-to-Text systems (STT), also known as automatic speech recognition systems (ASR), are another crucial part of the speech processing field and a tool to reduce the gap between human and computer communication methods. Its goal is to transcribe human speech from an input waveform (Rista and Kadriu, 2020). This process can also be understood as the speech synthesis process in reverse: we are searching for a way to convert spoken speech to a textual form. The difficulty of this task not only resides in the enormous diversity of human voices that need to be understood but also in dealing with the environment, which can be noisy and variable.

Speech recognition has a wide variety of applications, like human-computer interaction, acting as a bridge for computers to understand what humans communicate. In this sense, speech would be a much more straightforward and convenient interface than a keyboard, for example, when hands are busy with another part of the task to accomplish. As with speech synthesis, speech recognition is helpful in conversation agents and navigation systems or as a simple and fast way to send instructions to the application. Another possible utility of these tools is a simple audio transcription for optimal information storage or faster transmission. At the same time, it can be used to help people who have difficulties using a keyboard, mouse, or any other computer interface, either because they have a disability or because they have trouble adapting and communicating comfortably with these new systems.

1.3 Machine Translation (MT) Models

Recent Machine Translation systems, also known as Neural Machine Translation systems (NMT), are a crucial part of text communication in today's globalized world (Yang et al., 2020). Their main purpose is to translate text from one language to another without human interaction. This can facilitate and enrich the digital communication of two people who speak different languages.

Depending on the languages involved and the models available, the language translation can be done directly, or it may require multiple steps. For example, to translate a text from Basque to Spanish, one can use a Basque-to-Spanish translation model if such a model exists. If that is not the case, a possible approach is to use two models, translating to an intermediate widely available language in between, like English. In other words, translate Basque to English first and then translate the English text to Spanish. Besides, Machine Translation models with multi-language support for the source or the target language also exist, allowing to have an input or output in different languages. Therefore, in language translation, multi approaches are usually possible to translate a given text successfully.

1.4 Motivation and Scope of the Thesis

There is no doubt that Text-to-Speech, Speech-to-Text, Machine-Translation, and other speech and natural language-related technologies are gaining increasing importance as our social life extends into the digital realm, some becoming the key component for people to communicate satisfactorily. Due to this, the demand for this type of technology is skyrocketing, its use is spreading unstoppable around the world, and much research in this regard is developing and flourishing. This has led to significant advances in recent years, both from an academic and business point of view, from achieving multi-language support to creating multi-speaker systems that preserve the voice or imitate human prosody.

As each one of the parts of a Speech-to-Speech system has a considerable complexity, the main focus here will be to develop a system capable of synthesizing text following the latest trends in state of the art. This Text-to-Speech system will support multiple speakers and even zero-shot speaker synthesis. Until now, many of these researches focus on the English language. Here we will focus on developing Text-to-Speech models that support languages spoken in Spain, including but not limited to Basque and Spanish. Early versions of the transcription (STT) and translation (MT) modules will also be developed, being the result of initial research. Still, their more detailed development and improvement will remain pending for future research.

Typically, the training of these types of models requires massive audio datasets and text corpora. For English, there are many resources available online and for free. But for other languages, its availability is much more reduced, and often it requires a more exhaustive manual review and clean-up. In this research, the process followed for obtaining and pre-processing the data to be used will also be detailed. On top of that, during the development of this research, at some points, it has been necessary to use computers with high-end GPUs specifically designed for deep learning. When necessary, the hardware used and the approximate time required to create the models will be detailed.

1.5 Research Questions

1. Which are the correct procedures to develop and evaluate a Text-to-Speech system that supports multiple languages like Basque and Spanish?

2. Does adding languages to a state-of-the-art multilingual synthesis model decrease its performance?

3. Can these synthesis models be successfully incorporated into a Speech-to-Speech model?

1.6 Outline of the Master's Thesis

This thesis describes as detailed as possible the research process carried out. Starting from the study of the different initial investigation paths available, the decisions made, and the process followed to develop and evaluate the models. On the journey, the difficulties encountered are described, the strategies designed to overcome them are presented, and some limitations will be explained.

The work is organized as follows:

1. Chapter 2 provides an overview of the state-of-the-art Text-to-Speech technologies, focused mainly on the recent trends, including the models finally chosen. The needs and reasons for reaching the election will be explained.

2. Chapter 3 explains the methodology used to train and evaluate the Text-to-Speech models. This includes the analysis and pre-processing of the datasets.

3. Chapter 4 will show the results obtained when evaluating the models, making a thorough comparison between the different models trained and other state-of-the-art models.

4. Chapter 5 will present a complete Application that uses our Text-to-Speech models to do Speech-to-Speech Translation with voice conversion. In this chapter, the MT and STT models used will be introduced, including a section about how to train two of the STT models.

5. Chapter 6 will complete the research giving a summary of what has been achieved, with the conclusions reached and future research steps.

2 Literature Review

For the reader to acquire the required background knowledge to understand how the models used here work and the scope of our contribution, some of the traditional technologies involved in developing speech synthesis will be explained first. Afterward, an overall introduction to the latest models will be provided, explaining their strengths and weaknesses. The selected models for the work carried out here will be justified along the way. In order for the reader to understand this section properly, even though not strictly required, some previous experience with speech technologies and neural networks is recommended.

2.1 History

The interest in building devices in the shape of a human head capable of speaking dates back to the year 1003, with Gerbert of Aurillac (Pope Sylvester II) (R. M. Thomson, 1999). The stories tell that back then, they had already built a head capable of answering yes or no to any question. From there on, multiple stories were written around mysterious machines capable of speaking and with immense wisdom (LaGrandeur, 1999). Leaving aside the reliability of these stories, the interest of human beings in having entities or machines capable of speaking is indisputable. Without going that far in time, one of the first machines capable of synthesizing voice, of which there is real evidence, dates back to 1779, when Christian Gottlieb Kratzenstein built a tract-shaped device that could pronounce five vowels (Sami Lemmetty, 1999). Shortly after, Wolfgang von Kempelen described a design with mechanical equivalents of many parts of human vocal apparatus like the lungs, the glottis, mouth-shaped cavities, and machinery to control the movement of the lips, nostrils, and tongue-palate (Kempelen, 1791). A machine with this design should be elaborate enough and manage to pronounce some easy words and short sentences in an understandable way (Dudley and Tarnóczy, 1949). Based on this design, multiple speaking machines were built during the 19th and 20th centuries (Mattingly, 1974).

Regarding the use of computers for speech synthesis, the first attempts came around 1938, when Homer Dudley (Paul, 2014) developed the vocoder at Bell Labs. This hardware created tried to mimic speech based on a parametric method to produce tones and resonances. In the early days, including the twentieth century and the early years of this last century, the main speech synthesis techniques were articulatory, formant, and concatenative. More recently, statistical techniques gained more attention, in which the machines already consisted of training methods. All this until recent times, when neural networks have made their way in multiple fields, especially after the progress caused by attention and transformers in 2017 (Tan et al., 2021). In Image 3[1], we can see a summary of the most prominent techniques throughout each decade. The first model types were more parametric and knowledge-driven, requiring fewer data. The next models, being more concatenative, tend to require more data. The latest models, having a neuronal approach, are a mixture of both previous methods, being also parametric, but those parametrizations depend

[1] Updated version of the image obtained from Speech Technologies course at University of the Basque Country UPV/EHU.

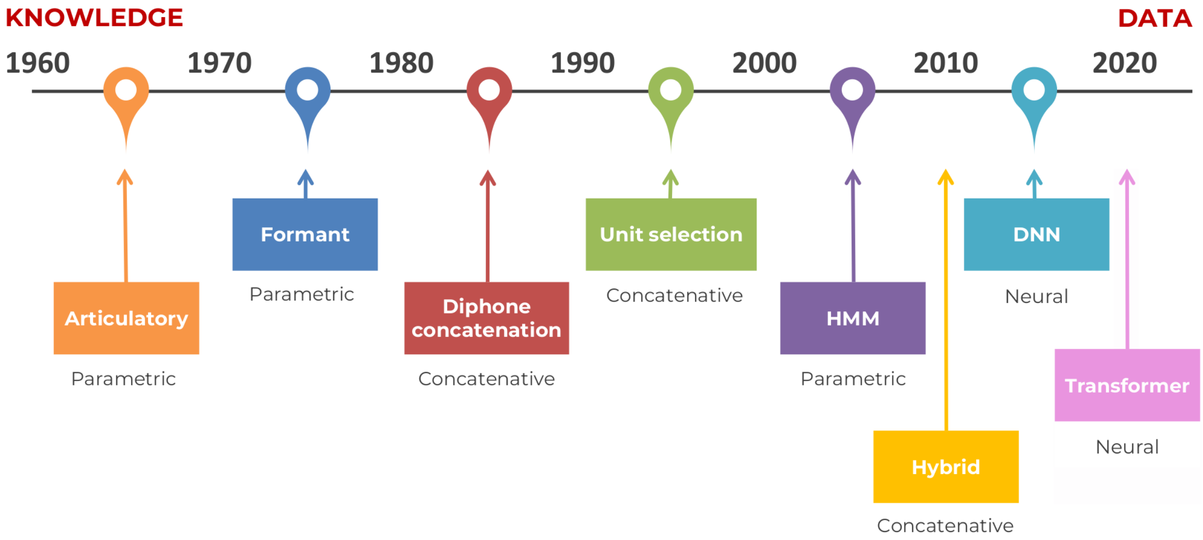


Figure 3: Computation-based approaches to speech synthesis throughout history.

on heavy data instead of human experts. Now, each of these computer-based synthesis methods will be briefly explained.

2.1.1 Articulatory Synthesis

Articulatory Synthesis methods were the first to appear and consisted of simulating vocal system parts like the vocal tract, lungs, tongue, and lips. This simple method of learning by imitation sounds simple at first, but in practice, with these techniques is incredibly difficult to imitate the sound of humans with precision. This is mainly due to the complexity of the human body and the wide variety of movements and articulations involved during speech. Among other reasons, being able to collect detailed information about the external and internal movements of our speech system becomes a daunting task. Consequently, articulatory approaches never quite worked well (Coker, 1976; Shadle and Damper, 2001).

2.1.2 Formant Synthesis

Formant Synthesis methods came next to address the problems of articulatory models. These model parameters are a set of rules to imitate the speech formants using digital filters. Usually, three to five formants are considered, depending on the required quality. This method allows the creation of an infinite number of sounds, where the quality of speech can be specified. Other configurable parameters are voicing fundamental frequency, excitation open quotient, voicing in excitation, and noise. However, the rules are challenging to define, and deep linguistic knowledge is usually required. In other words, these rules are parameters that need to be determined to produce the desired utterance and are difficult to create. Indeed, this system can create very intelligible speech using low computing resources and does not need a large dataset. Embedded systems like toys, arcade games, and some synthesis-integrated chips in the eighties commonly used them. Nevertheless, the speech produced by these methods lacks naturalness, sounds very robotic, and does not resemble the human voice, often including many artifacts (Seeviour et al., 1976; Allen et al., 1979; Klatt, 1980, 1987).

2.1.3 Concatenative Synthesis

The Concatenative Synthesis technique is just a concatenation of previously recorded sentences, words, or syllables. As expected, it requires a huge dataset to store all the recordings that can cover a wide range of possible language sentences. Additionally, a unit selection algorithm is used to search for the most appropriate recorded speech unit for each introduced text. As the produced speech uses pre-recorded speech pieces directly, the final results are very intelligible. They can also sound more human-like but are limited to the recorded speaker and speech units. Naturalness is very dependent on the unit selection algorithm and the available recordings. If the dataset is very exhaustive and the algorithm chooses the best pieces, longer recording units will be used, and the result may sound highly natural. But when sorted units are concatenated, that naturalness can be lost in the blink of an eye. Additionally, sometimes the concatenation can produce changes in tone, prosody, or the transmitted emotion, resulting in even less natural results (Olive, 1977; Moulines and Charpentier, 1990; Sagisaka et al., 1992; Hunt and Black, 1996; Black et al., 2001).

2.1.4 Statistical Parametric Speech Synthesis

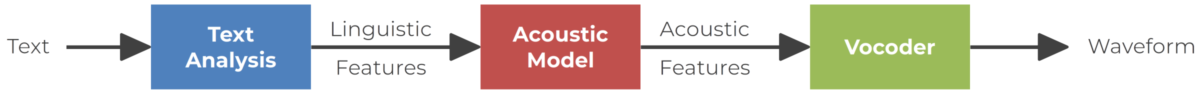


Figure 4: Statistical Parametric Synthesis speech synthesis process.

The next research field in speech synthesis was Statistical Parametric Speech Synthesis (SPSS). These models generate acoustic parameters, which are then used to generate the speech waveform. In these systems, the frequency spectrum (vocal tract), fundamental frequency (voicing), and duration (rhythm) can be modeled at the same time. The architecture of these systems is usually composed of three components: the text analysis module, the parameter prediction module, and the vocoder module. During synthesis, the text analysis module extracts the linguistic features, for example normalizing the text, extracting the phonemes, or a similar input linguistic transformation. These linguistic features are the input to the acoustic model. This acoustic model is often a Hidden Markov model and extracts acoustic features like fundamental frequency and spectrum simultaneously (Yoshimura et al., 1999; Yoshimura, 2002). Finally, the acoustic features go through a vocoder that converts them to a speech waveform. This speech synthesis process can be seen in Figure 4. For training SPSS systems, a dataset with paired linguistic features and acoustic features is required that will be used to train the acoustic model. The strength of these statistical systems is the naturalness of the audio and the flexibility compared to previous systems. In addition, the dataset does not need to be as big as for concatenative systems. However, as for the quality, albeit the speech produced tends to be intelligible, it often has many artifacts and sounds robotic. For that reason, it has a certain lack of naturalness, and the vocoder limits its quality to some extent. To overcome some of these limitations, the first neural network-based speech synthesis solutions replaced the HMM with a deep neural network (DNN) (Zen et al., 2013) or a recurrent neural network (RNN) (Fan et al., 2014). However, they still predicted acoustic features from linguistic features, maintaining a separate text analysis and vocoder module.

2.1.5 Neural Models

The latest trends in speech synthesis are Neural Speech Synthesis models. These systems continue the SPSS modular approach but replace each of the parts with their neural counterparts or sometimes even integrate them in a unified system, what is known as the End-to-End Neural Speech Synthesis model. The neural models use Deep Neural Networks (Gegout et al., 1995) based on an old multilayer perceptron architecture (Haykin, 1994). The perceptron is a computer-based neuron that, although it does not look like the neurons of the human brain, it was inspired by them and has the capacity to learn. A single perceptron is a function to map an input to a desired output using an approximate function. The weights are updated to get the actual output to match the desired output as they pass through the different examples from the dataset. This weight adjustment process is what is called training or the learning process of the neural network. Equation 1, is the expression used to calculate the output from the input. The o refers to the output, x_k to the input of the k artificial neuron, and W_k to its learnable weights. The first step is to get the weighted sum of the inputs, which goes through the function $\sigma(\cdot)$, known as the step function. This last function is usually a nonlinear activation function responsible for breaking the proportionality between the input and output.

Nonlinear problems are frequent in engineering, physics, biology, and other fields because most systems in nature are nonlinear. According to the Universal approximation theorem (Hornik et al., 1989; Csaji, 2001), putting together multiple nonlinear perceptrons, formed each by the weighted sum and the activation function, will allow us to approximate arbitrarily complex functions. The $\sigma(\cdot)$ function in the equation is also known as the Sigmoid activation function, and it is the most frequently used one, but others like Tanh, ReLU, or LeakReLU are also becoming common, especially for bigger networks Gustineli (2022). In Figure 5, we can see the general schema of a traditional perceptron.

$$o = \sigma\left(\sum_{k=0}^{n} x_k \cdot W_k\right) \text{ for } x_0 = 1 \tag{1}$$

The **Feed-Forward Neural Networks (FFN)** or Multilayer Perceptron (MLP) used by deep learning models are based on stacking multiple layers of perceptrons together.

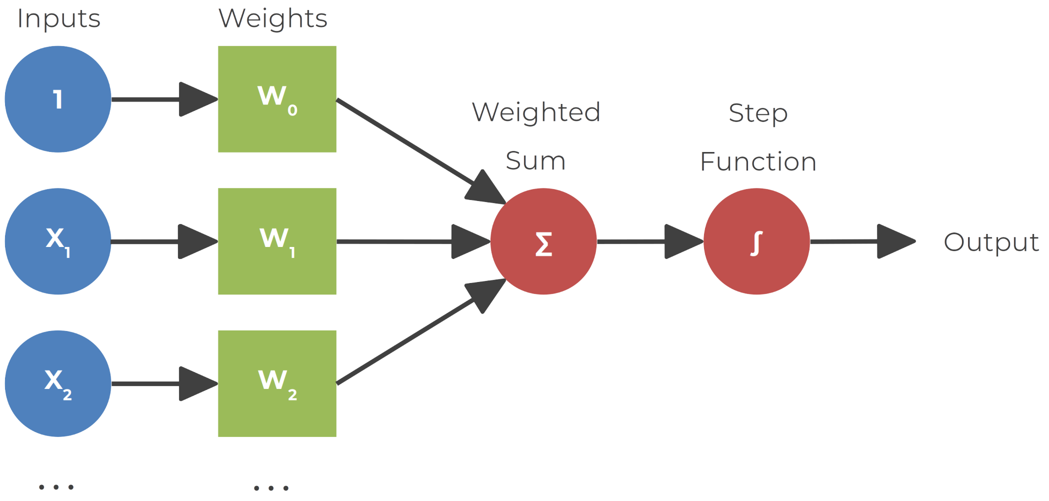


Figure 5: Single perceptron architecture.

Each layer in the middle has its own weights and might also have its own activation functions. These layers' architecture is composed of the input layer, an output layer, and hidden layers in-between with the perceptrons. To train these possibly big or huge neural networks efficiently, adjusting all the weights as it goes through all the examples in the dataset, multiple algorithms are available, being backpropagation (Rumelhart et al., 1986) with Stochastic Gradient Descent (SGD) (Robbins, 1951) the most common. The final architectures may vary in size and structure, but the most basic feed-forward network can be seen in Figure 6.

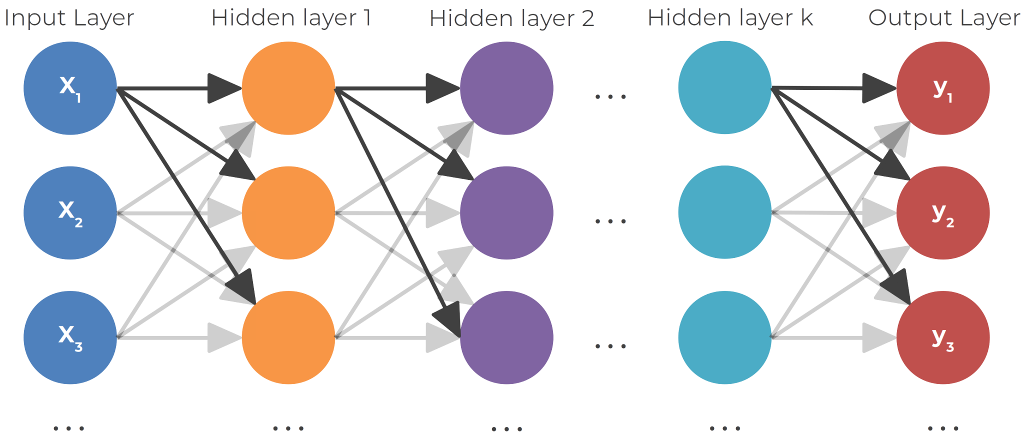


Figure 6: Feed-Forward Neural Network architecture, with multiple layers of perceptrons.

A variant of neural networks widely used in natural language processing and speech-related tasks are **Recurrent Neural Networks (RNN)** (Sherstinsky, 2020). The main idea of these networks is to use the previous output as input for the next step and are used to process sequential data: this type of model is also known as autoregressive. The weights of the different steps are shared across time in a hidden state. This allows part of the information from the past to propagate to the future, being appropriate to conserve some previous knowledge. Another use of these networks is for time-series data, like predicting future status based on previous data. There exist different possible implementations depending on how much or how we want previous information and merge it with the new input, the most common ones being Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014) networks. Depending on the task at hand, the recurrent neural networks can support different input-output configurations like one-to-many (i.e., text generation), many-to-one (text classification), and many-to-many (i.e., machine translation or speech synthesis). In Figure 7, there is a general diagram of a recurrent neural network in a many-to-many setup. The flexibility of these networks allows the creation of sequence-to-sequence (seq2seq) models with encoder-decoder modules (Sutskever et al., 2014), where there is a first RNN that encodes the input into a compressed internal representation, and then a decoder sub-model, that can be another RNN that can convert that internal representation into the desired output. In this internal representation, an attention sub-module can be attached to help the model learn the importance of past and-or future values (Liu and Lane, 2016). The main weak point of these networks is that they cannot run in parallel due to their structure, so they have some efficiency problems; for example, the inference time grows linearly with the output length. Additionally, their ability to retain older information is not so good. With respect to Speech Synthesis, the Tacotron model series (Wang et al., 2017; Shen et al., 2017) is the most known model to incorporate recurrent neural networks.

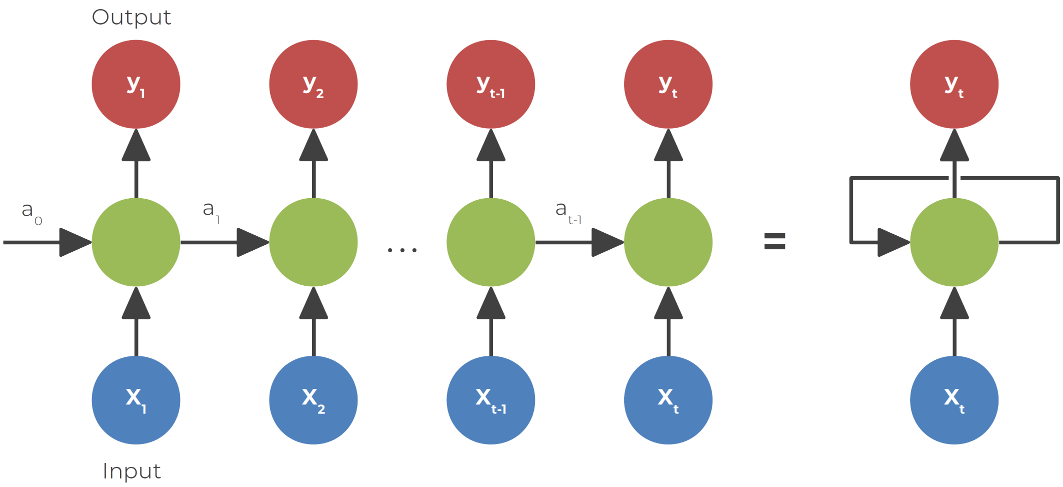


Figure 7: Recurrent Neural Network (RNN) architecture, in many-to-many kind of task.

Other widely used neural networks are **Convolutional Neural Networks (CNN)** (O'Shea and Nash, 2015), initially more focused on image-related tasks. The most important parts of these networks are convolution layers and pooling layers. The convolutional layer scans the input with a square sliding window and applies learned filters with weights as it traverses the image. Usually, the activation function used by these layers is ReLU instead of Sigmoid. The output of these layers is different feature maps, each enhancing a different characteristic of the image, like colors, edges, specific shapes, and so on. Typically, after each convolutional layer, there is a pooling layer that is just a downsampling operation on each feature map. Pooling operations are fast and simple, like getting the maximum or average value of a sliding window, and they reduce the dimensionality preserving the spatial invariance of the input image. In Figure 8, we can see a convolutional neural network with a single convolutional layer followed by a pooling layer. Frequently the models contain multiple convolutional and pooling layers in a row, sometimes intercalating another kind of neural network in between. The first layers tend to be good at recognizing small patterns of the images. As we go into the deeper convolutional layers, the neural network may gain the ability to recognize more complex shapes and more general aspects of the image, like objects or faces, depending on the task it has been trained on. Contrary to RRNs, CNNs can be processed in parallel to train these models faster. As more elaborate models have appeared, mixing several of these approaches, convolutional networks have made their way into tasks that are not only related to images. It is worth mentioning the case of 1D convolutions, where the window only slides along one dimension. This is appropriate for time-series data, where a sliding window can help extract features and maintain spatial information.

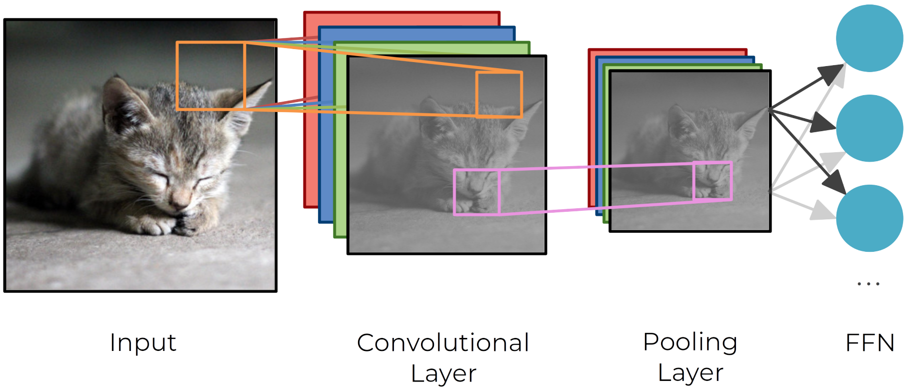


Figure 8: A Convolutional Neural Network (CNN) with a convolution and a pooling operation, with a feed-forward neural network at the end for the downstream task.

Nowadays, CNNs have been incorporated into the toolset to extract important features from diverse types of input. In the Speech Synthesis field, DeepVoice (Arik et al., 2017) uses a statistical model with convolutional neural networks, its latest version, Deep Voice 3 (Ping et al., 2017), becoming completely convolutional.

In 2017, a group of researchers was about to change the future of deep learning forever, releasing an article called ”Attention Is All You Need,” in which they presented a new model that they named **Transformer** (Vaswani et al., 2017). The task of the model was

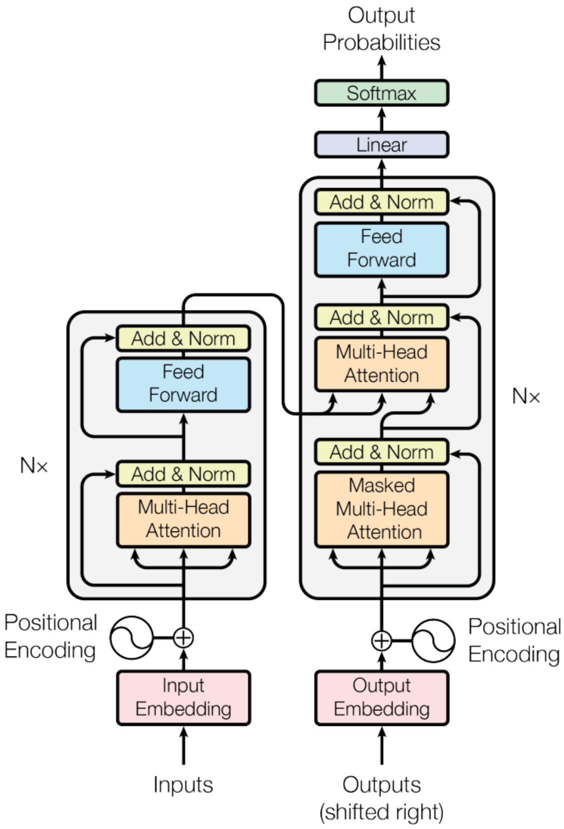


Figure 9: The transformer model architecture from the original paper (Vaswani et al., 2017).

translating text between languages, so both input and output are text. As for the architecture, it has an encoder-decoder structure, but instead of RNNs, the encoder and decoder modules were a list of layers composed of feed-forward networks, batch normalization, and self-attention modules. The general architecture of the transformer can be seen in Figure 9. One of the significant changes that the transformer brought about was being able to train in parallel, solving the previous problem with neural networks and scaling it in size without suffering from overfitting. In the speech field, Transformer TTS was one of the first models to incorporate transformers for synthesis (Li et al., 2018). Today its use is much more widespread.

With respect to the transformer's internal architecture, the batch normalization layers of the transformer (*Add & Norm*) just re-center and re-scale the data using the mean and variance to help the models train faster and more stable. The softmax function converts the output to a probability distribution of all the possible output values (Bridle, 1989). The self-attention modules learn the importance of different parts of the output generated by the previous networks, considering different criteria.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

The multi-head attention layer keeps attention to multiple parts of the sequence in different ways. It has eight attention heads running in parallel, each having three inputs called query (Q), keys (K), and values (V), as can be seen in Equation 2. The queries are the representation of the current word used to score against all the other words. We only care about the query of the token we are currently processing. Keys are like labels for all the words in the segment we are processing; in other words, what we match against in the search for relevant words. Values are the representations of the actual word; once we have scored how relevant each word is, these are the values we add to represent the current word internally in the model.

In Figure 10, we can see a diagram of the inner workings of one attention head. For a deeper understanding of this process, the reader may refer to "The Illustrated Transformer" article by Jay Alammar (Alammar, 2018).

So far, we have already seen the main pillars used to build the most recent neural models. The following section will mention specific speech models that are giving the best results today.

2.2 Recent Trends

The number of models for speech synthesis has had a huge increase in the last decade, even more so since the publication of the transformers and the greater availability of big datasets. Some of the models still use an external vocoder; others include the vocoder inside their architecture. The text analysis module is also greatly simplified, usually reduced to a simple text normalization or grapheme-to-phoneme conversion. In brief, the general trend is the creation of End-to-End models that do the synthesis task directly and depend less and less on external modules or previous steps to work.

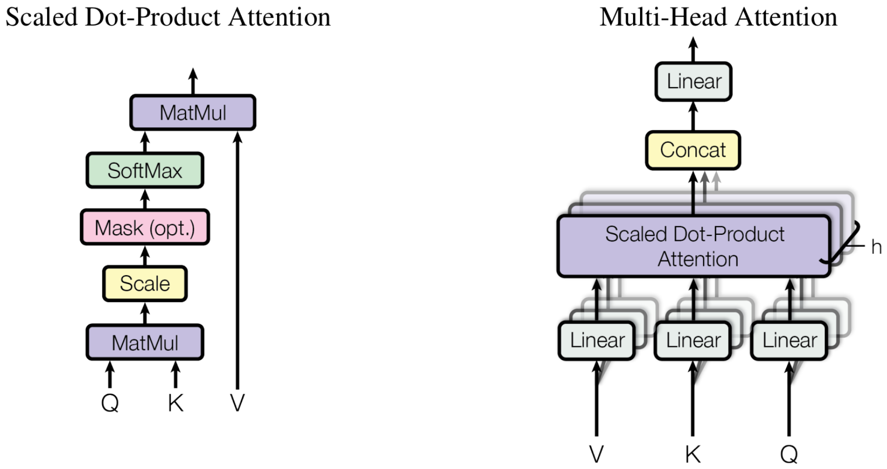


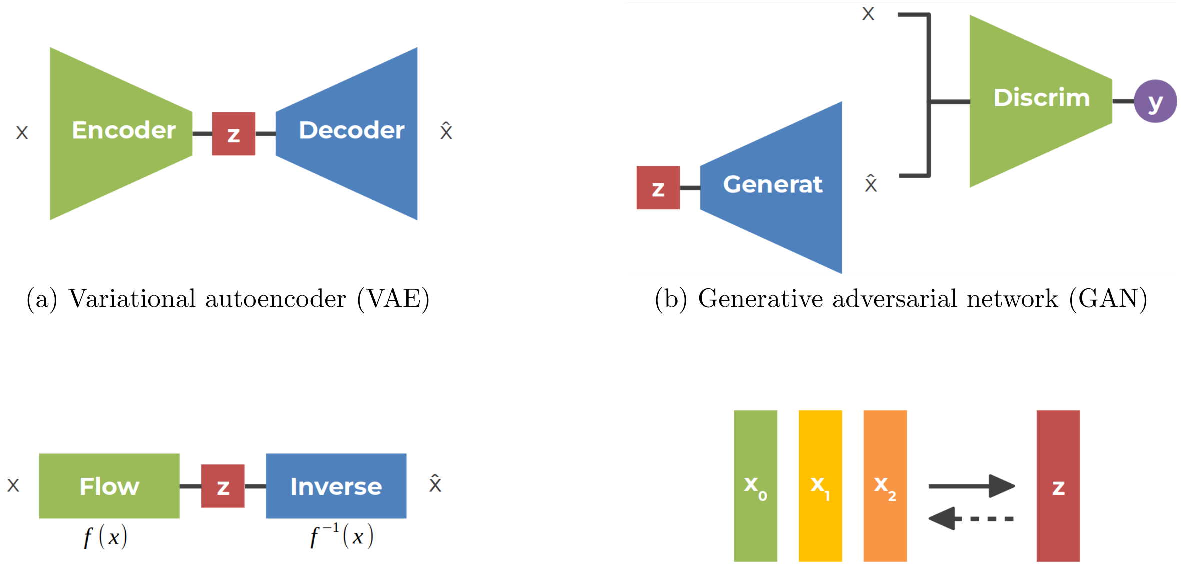
Figure 10: Transformer attention heads diagram from the original paper (Vaswani et al., 2017).

Before delving into specific models, a new trend in deep learning models should be introduced: Generative Modeling (Ng and Jordan, 2001). Many traditional machine learning models are discriminative in nature, where they try to model the output of the model knowing the input ($p(y|x)$). However, generative models try to model the joint probability of the input and the output instead ($p(x, y)$). In other words, they try to learn the probability distribution of the dataset, and we will be able to generate outputs stochastically just by sampling from the model. For example, in the image domain, generative models are able to create new pictures that do not exist in the original dataset, but they look like they come from it. This implies that these kinds of models are probabilistic instead of deterministic, and they will generate a different output each time during inference.

In Figure 11, the most known recent trends in generative modeling are listed with a simplified diagram of their architecture. Different generative techniques exist mainly to add complexity to the data and are commonly used for the one-to-many types of problems, like in text-to-speech conversion, in which, for simple text input, there may exist multiple valid speech waveforms that have more complexity.

Variable Autoencoders (VAEs) (Kingma and Welling, 2013) (Figure 11a) are generative models where the input is compressed into a lower dimensional space, generating a smaller vector that tries to conserve as much information as possible from the input (x), but in a smaller latent space. Then, from that generated z latent variable, the original input can be reconstructed ($\hat{x}$). The Encoder part usually includes convolutions or similar operations to reduce the input vector. The decoder is composed of convolutional transposed layers to increase the size of the latent variable back to the original. After training the model, by sampling variables from the latent space created, the model will gain the ability to generate new outputs that resemble the examples in the dataset but did not exist previously.

Another new generative approach is **Generative adversarial networks (GANs)**



(a) Variational autoencoder (VAE)

(b) Generative adversarial network (GAN)

(c) Flow-based models

(d) Diffusion models

Figure 11: Recent trends in generative neural models. Diagrams inspired by various Lilian Weng's articles (Weng, 2018a, 2017, 2018b, 2021).

(Goodfellow et al., 2014) (Figure 11b), which are not so focused on estimating the density of the distribution but more on sampling to generate new output. The sampling can be done from some simple noisy input, and the model will learn a transformation to the learning distribution. The GAN is composed of two networks that will compete with each other. There is a Discriminator network that will learn how to differentiate real examples on the dataset from generated examples. There is also a Generator network that turns the input noise into outputs that mimic training dataset examples and will try to trick the Discriminator. With this architecture, the discriminator network will teach the generator network how to create examples that are apparently real, and the Generator will teach the Discriminator how to classify the generated non-real examples correctly. Often these kinds of models are trained interleaved, first the discriminator for some steps, then the generator, later the discriminator again, and so on, but varied approaches exist. After training the GAN, the generator network can be used to generate new examples similar to the ones in the original dataset.

Flow-based models (Rezende and Mohamed, 2015) shown in Figure 11c learn the probability density function using a sequence of invertible transformations. The different transformations are closed under composition and are called one after the other. In this way, the model will be able to learn to generate complex flows from simpler ones. In the end, the model will learn a useful latent representation that has immediate mapping with the input. These models are efficient to sample and evaluate, giving highly expressive and flexible output, and also straightforward to train. For example, during training, these models can learn to create Gaussian noise from the output examples; during inference, the model will know how to generate examples just by sampling from Gaussian noise. The primary condition for the process to work is that the mapping process needs to be formed by diffeomorphism functions: in layman's terms, all the functions need to be both invertible and differentiable.

Diffusion Models (Sohl-Dickstein et al., 2015; Ho et al., 2020) diagram can be seen in Figure 11d. Diffusion Probabilistic Model (DPM) models create a Markov chain of steps to keep adding noise to the data while performing the reverse diffusion process. Diffuse model training consists of slowly incorporating random noise into the original data step by step, and the model learns to reverse the process little by little (Weng, 2021). In the early stages of training, the model will learn to remove a tiny amount of noise from the data. In later stages, the model will be able to create data samples from pure noise that closely resemble the original dataset examples. These kinds of models can be trained using a framework called stochastic calculus (Song et al., 2020).

Leaving the different approaches aside, and with respect to specific recent models, in Table 1, we can see the best-performing TTS models at the end of 2022. The training of these models was made by the NaturalSpeech (Tan et al., 2022) project researchers, retraining all the models using the same dataset and splits. Here we are going to give an overview of those best-scoring speech synthesis systems, introducing some concepts along the way.

Model	**Type**	**MOS**	**CMOS**	**Time**
FastSpeech 2	Self-Attention	4.32 ± 0.15	-0.33	2020
Glow-TTS + HiFiGAN	Flow	4.34 ± 0.13	-0.26	2020
Grad-TTS + HiFiGAN	Diffusion	4.37 ± 0.13	-0.24	2021
VITS	Non-autoregressive	4.43 ± 0.13	-0.20	2021
NaturalSpeech	Self-Attention	4.56 ± 0.13	0.00	2022

Table 1: Test results of the latest TTS systems on the LJSpeech dataset.

2.2.1 WaveNet

Before explaining the presented state-of-the-art models, let us see an overview of a WaveNet model. WaveNet was presented in 2016 by the Google DeepMind team as an autoregressive model using convolutional neural networks to generate raw audio waveforms directly (Oord et al., 2016a). It was proposed more as a polyvalent framework. and it stood out among other proposals in that it can be very versatile: it can be applied to Text-to-Speech tasks, as a vocoder, or even as a discriminative model for phoneme recognition and similar tasks. The brought innovations were its naturalness when evaluated subjectively, ability to handle long-time dependencies, being able to condition different aspects like specific speakers, and support of different kinds of waveform outputs not only limited to human speech. In this model, dilated causal convolutions are used to achieve long-time dependency support, and they are composed of a stack of convolutional layers. These layers cannot depend on the future but can depend on the long-time past by using convolutions with holes. Those convolutions skip certain input steps to increase the perceptive temporal field of the model.

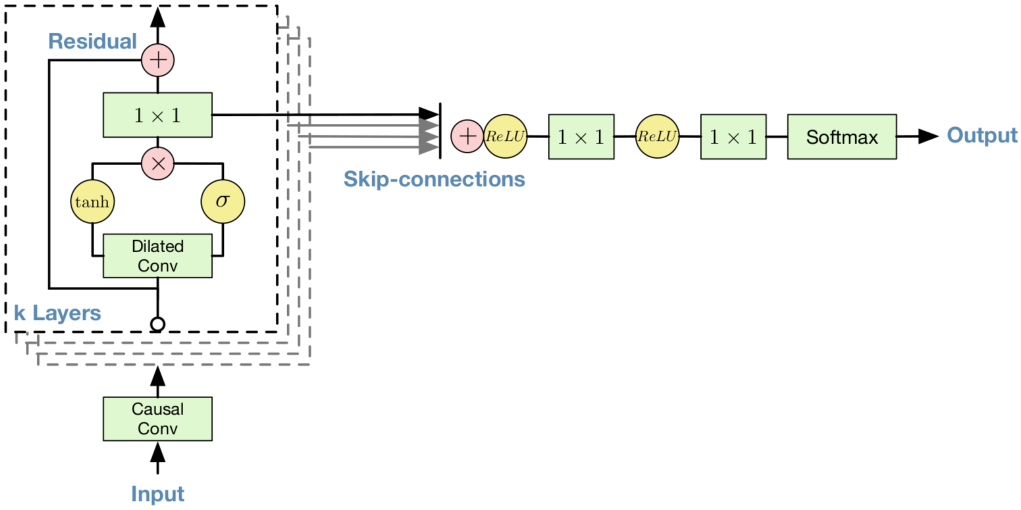


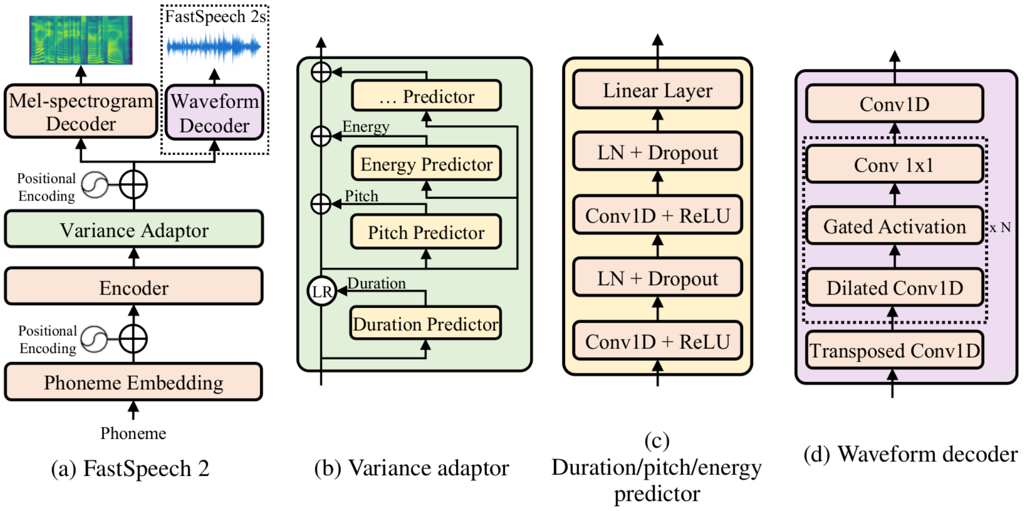
Figure 12: WaveNet model architecture from the original paper (Oord et al., 2016a).

In Figure 12, the general diagram of the residual blocks that form the WaveNet can be appreciated. Residual block means that the layer can be skipped, by merging the output of the module with the input, to have a view of short-term and long-term at the same time. After each dilated convolution, the use of gated activation units can be seen as in PixelCNN (Oord et al., 2016b). In Equation 3, the operations performed by this activation function are presented, where $\odot$ means element-wise multiplication, $*$ denotes convolution, f refers to filter, g to gate, and W are convolution filters to be learned during training.

$$\mathrm{z} = \tanh(W_{f,k} * \mathrm{x}) \odot \sigma(W_{g,k} * \mathrm{x}) \tag{3}$$

2.2.2 FastSpeech 2

FastSpeech 2 (Ren et al., 2020) is a text-to-speech synthesis model developed by researchers at Microsoft. FastSpeech models are non-autoregressive in the sense that they replaced the traditional RNNs with FFNs. This allows them to overcome some of the known limitations of RNN models, mainly their slowness due to the difficulties of training them in parallel and efficient use of GPUs; also, their lack of robustness results in frequent word skipping, mispronunciation, and repetitions (Chen et al., 2020b; Peng et al., 2019). At the same time, FastSpeech models are able to achieve similar voice quality to traditional autoregressive models. On the contrary, unlike autoregressive models, they cannot have infinite memory, so they usually need to be bigger to process longer inputs. FastSpeech 2 is an improved version of the original FastSpeech 1 (Ren et al., 2019) model that generates high-quality speech at a faster rate. For that, it adds a variance adapter module to their previous version and directly trains the model with ground truth examples.



(a) FastSpeech 2 (b) Variance adaptor (c) Duration/pitch/energy predictor (d) Waveform decoder

Figure 13: FastSpeech 2 model architecture from the original paper (Vaswani et al., 2017).

Therefore, FastSpeech 2 proposes to use a non-autoregressive approach to tackle the one-to-many problem of generating audio from text or phones. This is because, in the TTS task, the input lacks much information, like pitch, energy, volume, or prosody, that need to be generated by the model in order for the output to sound natural. For that, a variance adaptor is added that will learn to predict speech, energy, and phoneme duration during training and add this information to the input during inference. Phoneme duration information is important to determine the length of generated speech sounds: they use Montreal forced alignment (MFA) (McAuliffe et al., 2017) to extract the phoneme duration for training. Pitch is essential to express emotions and affects the prosody: they use Continuous Wavelet Transform (CWT) to extract the pitch spectrogram (Suni et al., 2013; Grossmann and Morlet, 1984) using it as the target. Similarly, energy is an essential feature for prosody and also affects volume: they use the amplitude of each Short-Time Fourier Transform (STFT) frame as the target. Hence, they propose using three predictor modules, each of which comprises a 1D convolutional network with ReLU activation, a normalization layer, and a dropout layer. The normalization re-centers and re-scales the data; the dropout omits some random neurons during training to avoid overfitting, and 1D convolution extracts spatial properties of speech time-series data. In Figure 13, we can see the model architecture. The encoder and Mel-spectrogram decoder layer are models based on the transformer architecture used in FastSpeech 1. In Figure 14, we can see the internals of the FastSpeech 2 encoder and decoder modules, similar to the original transformer blocks, replacing the FFN with a 1D convolution.

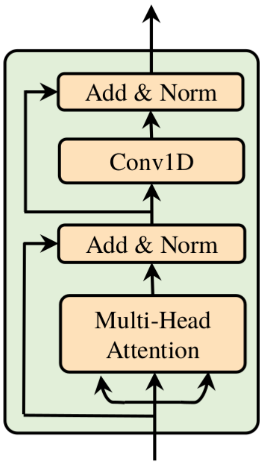
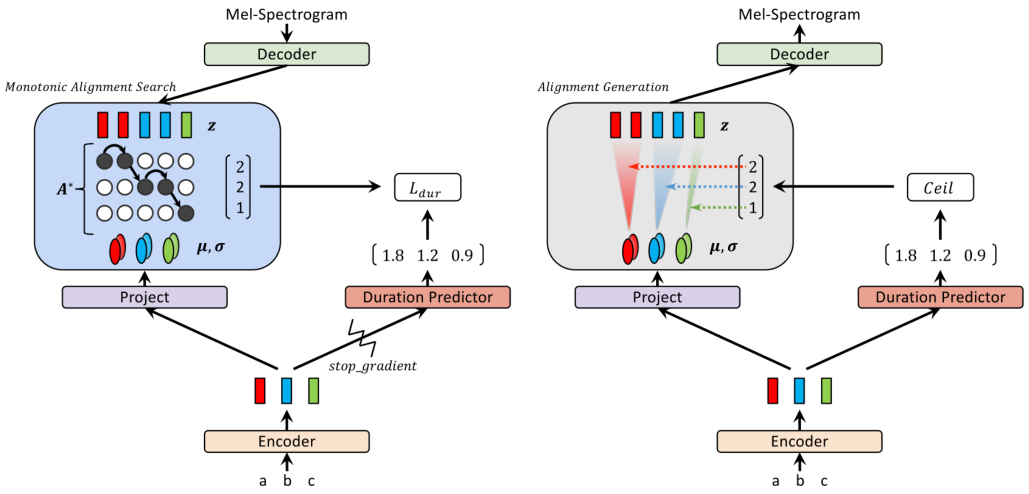


Figure 14: FastSpeech 1 FFT Block architecture from the original paper (Vaswani et al., 2017).

2.2.3 Glow-TTS

Glow-TTS (Kim et al., 2020) is a generative flow-based TTS model that is faster than autoregressive models but maintains good quality and gains in flexibility (Weng, 2018b). Flow-based models try to learn the probability density function of the data. For that, they use a statistics tool called normalizing flows (Rezende and Mohamed, 2015) that estimates the probability density function using a sequence of invertible transformations. This process makes it easy to calculate the loss and do the back-propagation making the estimation of probability density functions tractable. Compared with previous TTS models, Glow-TTS does not require an external aligner, so the training procedure is simplified. The model finds the alignments between text and speech using flows and dynamic programming, searching for the most probable monotonic alignment, also known as Monotonic Alignment Search (MAS). Duration predictor ensures monotonicity and surjectiveness; in other words, the characters are pronounced in the correct order, and no text is skipped or repeated. Using monotonic alignments also behaves better on long sentences and generally gets a more robust TTS. Using generative flows achieves a fast, diverse, and controllable synthesis. This control is performed by altering the intermediate latent representation and allows for tweaking intonation patterns and pitch. Additionally, the model is designed to be easily extended to a multi-speaker setup.

In Figure 15, we can see the general architecture and get an idea of how the training and inference procedures are performed on the Glow-TTS model. The Encoder module at the bottom of the figure is based on Transformer TTS (Li et al., 2018) encoder, a speech-oriented modification on the original Transformer encoder architecture (Vaswani et al., 2017). From the Transformer TTS, the positional encoding is replaced by a relative position representation. They also added a residual connection to the encoder before the neural network. Then a linear projection layer is added to estimate statistics of the



(a) An abstract diagram of the training procedure.

(b) An abstract diagram of the inference procedure.

Figure 15: Training and inference processes of Glow-TTS from the original paper (Kim et al., 2020).

prior distribution, referred to as *Project* in the diagram. With respect to the Duration Predictor, it is the same as in FastSpeech 1 (Ren et al., 2019) and FastSpeech 2 (Ren et al., 2020) (Section 2.2.2), with two convolutional layers using ReLU activation, a normalization layer, and dropout. The decoder module is a family of flows to do the forward and inverse transformation in parallel. It is a stack of multiple blocks with an activation normalization layer, a 1D convolution, and an affine decoupling layer based on WaveGlow model architecture (Prenger et al., 2018).

2.2.4 Grad-TTS

Grad-TTS (Popov et al., 2021) is a Diffusion Probabilistic Model (DPM) (Sohl-Dickstein et al., 2015) for speech synthesis. In short, Grad-TTS is a synthesizer using the monotonic alignment search (MAS) from Glow-TTS to align the encoder output; this is combined with a decoder that transforms Gaussian noise parametrized by previously aligned output into a mel-spectogram. A peculiarity of this model is that the user can control the balance between audio quality and inference speed. It is also possible to use this model in an End-to-End fashion, removing the vocoder by making it output a waveform.

In Figure 16, we can see the inference process and the model's internal architecture. The input to the model can be phonemes or characters, and the output will be mel-spectrograms, so a vocoder is needed to obtain the final waveforms. Internally, the Grad-TTS model contains three modules: encoder, duration predictor, and decoder. The encoder and duration predictor structure is the same as in the previously seen Glow-TTS model. The decoder is a Probabilistic Diffusion Model with a small version of U-Net architecture (Ronneberger et al., 2015; Ho et al., 2020) so as not to increase the size of the model too much. Besides, the model allows a variable number of steps of the decoder at inference to

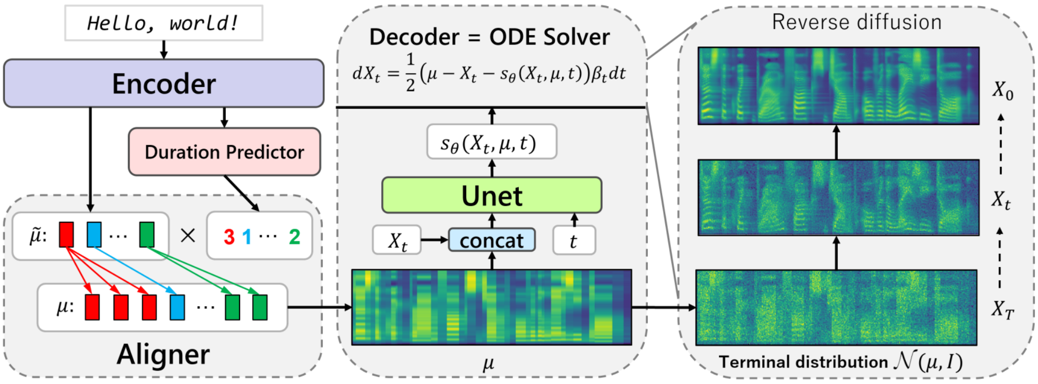


Figure 16: Inference processes of Grad-TTS from the original paper (Popov et al., 2021).

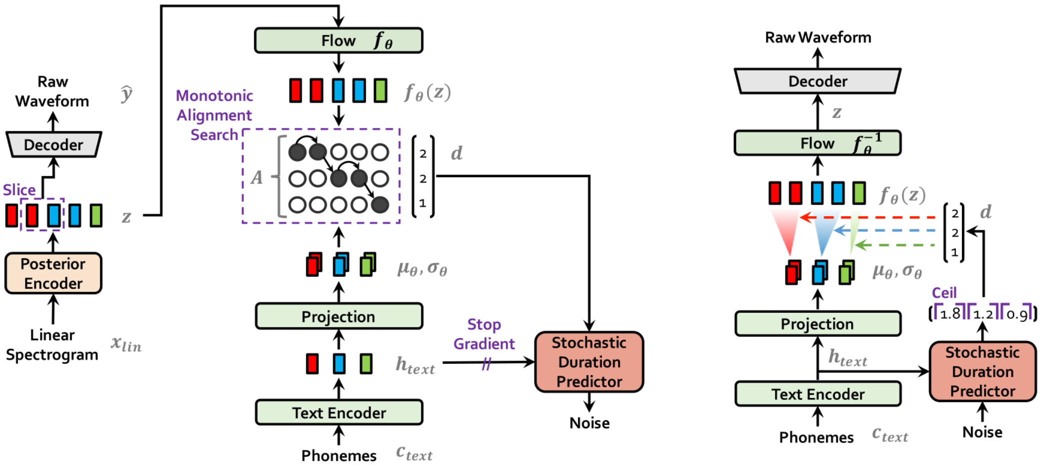
search for a better balance between speech quality and speed, being able to do real-time synthesis comparable to the quality level of other commonly used TTS.

2.2.5 VITS

Variational Inference with adversarial learning for end-to-end Text-to-Speech (VITS) (Kim et al., 2021) is a parallel end-to-end model that integrates the acoustic model and the vocoder in the same pipeline, generating a waveform from text or phonemes. Their paper introduces a new stochastic duration predictor module to add diverse rhythms to the speech, thus tackling the one-to-many problem. Additionally, it combines various recent techniques like normalizing flows, adversarial training, and a variable autoencoder. Normalizing flows improve expressiveness and generate high-quality waveforms. Adversarial training is used for waveform generation. Variable autoencoders are mainly to connect the acoustic and vocoder modules of the TTS. Let us dive into the details of this model for a moment to try to understand it.

In Figure 17, we can see the differences between the training and inference processes of the model. At the same time, we can get an overview of all the modules that compose it. When training the model, we have linear spectrograms and phonemes as input and the final waveform as output. During this process, the z latent variables will learn to have the required information from the aligned text and the spectrogram, facilitating the connection between the acoustic model and the vocoder. Even though not included in the schema and not studied in detail, they also demonstrate the expressive characteristics of the model by proposing a way to add a speaker embedding to multiple parts of the model.

From the diagram, the Posterior Encoder is the same as the encoder from GlowTTS: a WaveNet residual block with convolutional layers, a gated activation unit, and skip connection (Oord et al., 2016a). The Prior Encoder is formed by a Text Encoder to process the input phonemes and a Flow to add complexity to the prior distribution. The text encoder is a transformer encoder block using absolute positioning (Vaswani et al., 2017). The Flow is a normalizing flow layer to add flexibility, and it is a stack of volume-



(a) Training procedure

(b) Inference procedure

Figure 17: Training and inference processes of VITS from the original paper (Kim et al., 2021).

preserving affine coupling layers (Dinh et al., 2016) with a stack of WaveNet residual blocks. The Decoder is a HiFi-GAN V1 (Kong et al., 2020) vocoder. The vocoder has a Generative Adversarial Network (GAN) architecture (Goodfellow et al., 2014), this means that it has a Generator learning to create waveforms and a Discriminator learning to detect those fake waveforms; both models will compete with each other, ending up with a Generator able to output natural human-like speech. The Stochastic Duration Predictor is responsible for learning phoneme duration. It is a stack of residual blocks with convolutional layers. They apply neural spline flows with invertible nonlinear transformation (Durkan et al., 2019) to improve expressiveness similar to the affine coupling layers of the Flow module.

In order to successfully train all these blocks, the VITs model uses several losses together. In Equation 4, the combination of the VAE and GAN training final loss is presented.

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv}\left(G\right) + L_{fm}\left(G\right) \tag{4}$$

The mel-spectrogram Reconstruction Loss is the maximum likelihood estimation assuming a Laplace distribution between the mel-spectrograms. The real mel-spectrogram is compared with the mel-spectrogram of a wave generated by the vocoder when upsampling the latent variable z. The mel-spectrograms are calculated with the Short-time Fourier Transform (STFT) and a linear projection into the mel-scale. Then the L_1 loss between both spectrograms is calculated. The final formulation is in Equation 5.

$$L_{recon} = \|x_{mel} - \hat{x}_{mel}\|_1 \tag{5}$$

The Kullback–Leibler (KL) divergence loss in Equation 6 compares how much entropy there is between the input linear spectrogram (x_{lin}) and input character phonemes (c_{text}) with the estimated alignment (A), after applying the normalizing flow (p_θ). The KL divergence measures the statistical distance or how different both distributions are. Using the linear-scale spectrogram of the ground truth speech instead of the mel-spectrogram helps provide higher-resolution information to the posterior encoder.

$$\begin{aligned} L_{kl} &= \log q_\phi\left(z \mid x_{lin}\right) - \log p_\theta\left(z \mid c_{text}, A\right), \\ z \sim q_\phi\left(z \mid x_{lin}\right) &= \mathcal{N}\left(z; \mu_\phi(x_{lin}), \sigma_\phi(x_{lin})\right), \\ p_\theta &= \mathcal{N}\left(f_\theta(z); \mu_\theta(c), \sigma(c)\right) \left|\det \frac{\partial f_\theta(z)}{\partial z}\right|, \\ c &= [c_{text}, A], \end{aligned} \tag{6}$$

The alignment (A) is estimated with the monotonic alignment search (MAS) like in Glow-TTS (Kim et al., 2020), which maximizes the likelihood of data parametrized by a normalized flow (f), as in Equation 7.

$$A = \underset{\hat{A}}{\operatorname{argmax}} \log p(x \mid c_{text}, \hat{A}) \tag{7}$$

The Duration Loss (L_{dur}) is the negative variational lower bound (VLB) of the phoneme duration shown in Equation 8. The stochastic duration predictor is a flow-based generative model trained with maximum likelihood estimation. To solve the problem, they add a random variable (u) for variational dequantization (Ho et al., 2019) and another random variable (v) to apply variational data augmentation (Chen et al., 2020a), both of them having the exact same time resolution and dimension as the duration sequence (d). The phoneme duration is sampled from random noise using inverse transformation of the duration predictor.

$$L_{dur} = -\mathbb{E}_{q_\theta(u,v \mid d,c_{text})}\left[\log \frac{p_\theta(d-u, v \mid c_{text})}{q_\theta(u, v \mid d, c_{text})}\right] \tag{8}$$

For the adversarial training of the HiFi-GAN vocoder, a discriminator needs to learn to differentiate between generated examples and real ground truth examples. The GAN is trained jointly in a minimax game, where the discriminator wants to maximize the difference between the real data classification ($D(x)$) and fake data ($D(G(z))$). For that, traditional binary cross-entropy loss of the original GAN (Goodfellow et al., 2014) and Least-Squares loss functions from LS-GAN models (Mao et al., 2016) are used to avoid the vanishing gradient flow. The discriminator tries to classify real examples as 1 ($D(x)$) and generated examples ($D(G(z))$) close to 0, maximizing their difference. The least squares GAN Discriminator Loss can be seen in Equation 9. Conversely, the generator loss wants to minimize the difference between real and fake data. For that, it wants to make the generated data ($D(G(z))$) close to 1, as the real data. The least squares GAN Generation Loss can be seen in Equation 10.

$$L_{adv}(D) = \mathbb{E}_{(y,z)}\left[(D(y)-1)^2 + (D(G(z)))^2\right] \tag{9}$$

$$L_{adv}(G) = \mathbb{E}_z\left[(D(G(z))-1)^2\right] \tag{10}$$

Finally, Equation 11 defines the Feature Matching Loss of the Hifi-GAN. It calculates the distance between a generated sample and a ground truth sample in every intermediate feature. This measures the difference in features of the discriminator in each layer, and it is known to help train the generator in speech synthesis models (Kumar et al., 2019). It is like a reconstruction loss but measured in the hidden layers of the discriminator, and it works as a replacement for the element-wise reconstruction loss used by VAE models (Larsen et al., 2015).

$$L_{fm}(G) = \mathbb{E}_{(y,z)}\left[\sum_{l=1}^{T} \frac{1}{N_t} \| D^l(y) - D^l(G(z)) \|_1 \right] \tag{11}$$

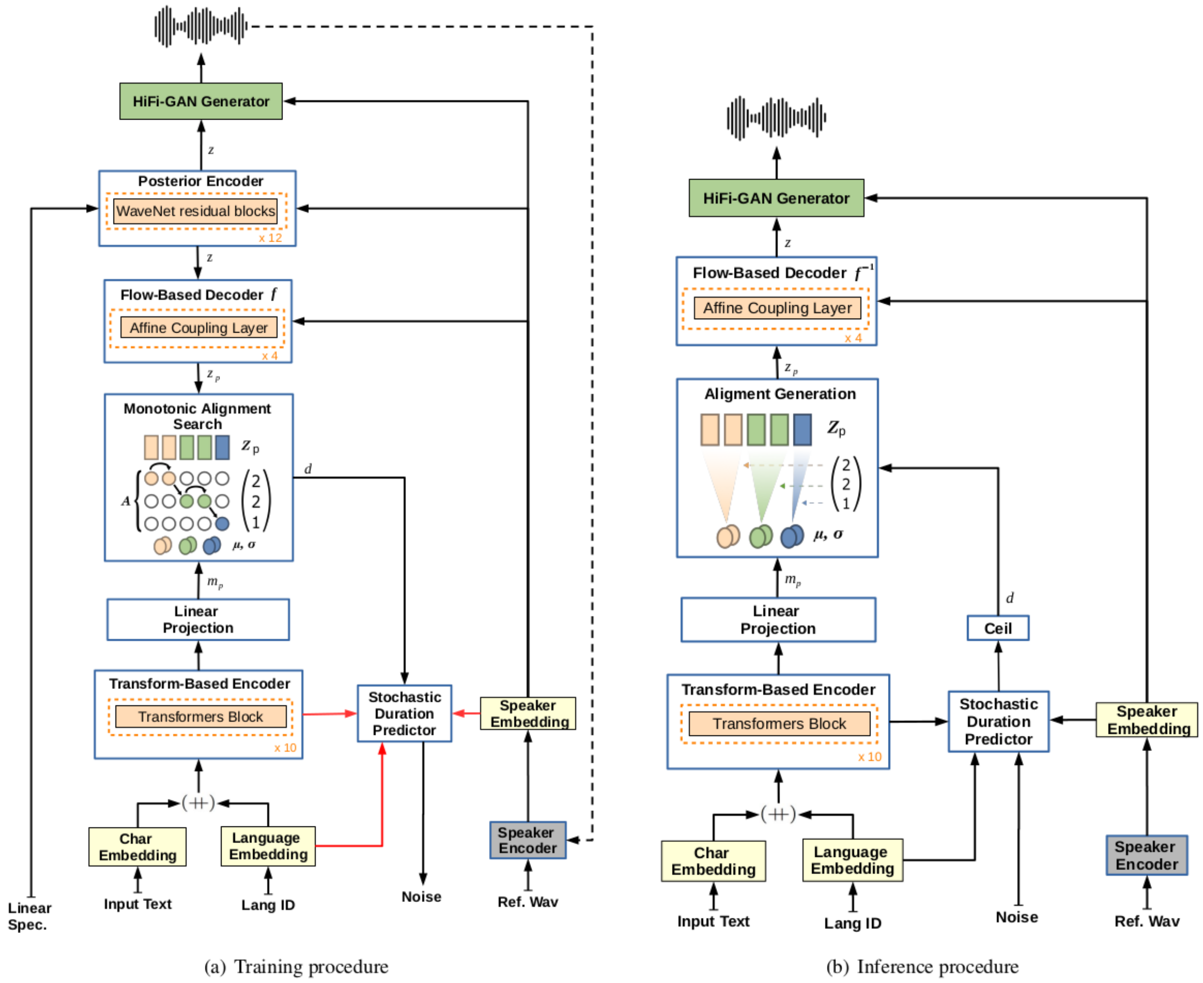
2.2.6 YourTTS

YourTTS (Casanova et al., 2022b) model is a multi-speaker and multilingual version of the VITS model. Both speaker and language are explicitly added information types through embeddings. It gets state-of-the-art (SOTA) results in zero-shot multi-speaker speech synthesis. Zero-shot multi-speaker is the ability of the model to synthesize voices that have not been seen during training: in other words, to conserve or imitate new speaker voices. It has been trained on four datasets: two English datasets, one French dataset, and a dataset in Portuguese. This last dataset contained only one speaker and was recorded with a non-professional microphone. This opens up a world of possibilities for languages with scarce resources. Usually, zero-shot models require a high amount of training data, with many speakers, and such a magnitude of resources are not available for most languages. Furthermore, the model can be fine-tuned in specific speakers with just around a minute of speech recordings to get better results.

In Figure 18, there is a detailed view of the model architecture. The red connections stop the propagation of the gradient during training, and the ++ character indicates concatenation. The main differences from the VITS model will be explained here. As with the VITS model, the Posterior Encoder is only used during training to glue the acoustic model with the vocoder, learning to create a z latent variable that extracts meaningful information from both the aligned text and the linear spectrogram. Additionally, the text input is not converted to phonemes, so the model can learn languages without a public grapheme-to-phoneme, as the conversion is not needed.

As for multi-language, the model supports three languages: English, Portuguese and French. A language embedding is added to the embeddings of the input characters. More languages could be added later using that embedding. Indeed, during their training process, they teach the model one language at a time, adding a dataset with a new language after being trained on the previous language.

With respect to multi-speaker support, during training, speaker embeddings are added to the flow-based decoder, the posterior encoder, and the HiFi-GAN vocoder. This method is also briefly introduced and tested in the VITS paper. They use global conditioning from WaveNet (Oord et al., 2016a) to influence the output distribution across all timesteps of the flow-based decoder and the posterior encoder. They also added the speaker to



(a) Training procedure

(b) Inference procedure

Figure 18: Training and inference processes of YourTTS from the original paper (Casanova et al., 2022b).

the transformer-based encoder and the decoder and passed it to the stochastic duration predictor. Moreover, they added Speaker Consistency Loss (SCL) (Xin et al., 2021) to the final loss. This loss maximizes the cosine similarity between the speaker embedding of the ground truth and the generated audio. We can see it in Equation 12, where the α value is a positive number to determine the importance of this loss in the final loss, and $\phi(\cdot)$ is the function returning the embedding.

$$L_{SCL} = \frac{-\alpha}{n} \cdot \sum_{i}^{n} \text{cos\_sim}\left(\phi(g_i), \phi(h_i)\right) \tag{12}$$

This Speaker Consistency Loss will be added to the final loss we showed before for the VITS model in Equation 4. The rest of the losses stayed the same, as we can see in Equation 13. The external H/ASP model is used to generate the speaker embeddings (Heo et al., 2020): this is the $\phi(\cdot)$ function above. This model is already trained and is not fine-tuned here. This speaker recognition model achieved state-of-the-art results in the Vox-Celeb 1 test split (Chung et al., 2018), a dataset containing more than 6,000 speakers.

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G) + L_{SCL} \tag{13}$$

2.2.7 NaturalSpeech

During the development of this research, a new promising model was released called NaturalSpeech (Tan et al., 2022) by a research team in Microsoft. This model far exceeded the results of the VITS and other state-of-the-art models, obtaining the best results in the art to date and without increasing inference time. As shown in the paper (Tan et al., 2022) and in Table 1 here, the quality of the generated audios has no statistically significant difference from ground truth recordings. In their paper, they also propose a definition of what human-level audio quality means in a statistical way using CMOS and Wilcoxon signed rank test (Wilcoxon, 1945).

In Figure 19, there is a general diagram of the model. As can be seen, it has some similarities with the previous models mentioned here. The inputs are phonemes, and the outputs are the raw waveform. The model also uses a VAE to reduce the speech signal into a compressed latent variable that conserves just the main information. The authors propose to use a VAE model with a memory bank with attention. The input to the Posterior Encoding is only used during training and are linear spectrograms like in VITS. The Posterior Encoder is also formed by WaveNet modules, in this case, 16 of them. The Wave Decoder is formed by 4 residual convolutions and represents the vocoder. The Bidirectional Prior/Posterior is a module based on flow models (Dinh et al., 2014) to improve the quality of the prior ($p(x|y)$) and simplify the posterior ($q(z|x)$). This is similar to the Flow modules seen before. The Phoneme Encoder has been previously pre-trained on an extensive corpus to create better representations. As in VITS, it is composed of Feed-Forward Transformers blocks from FastSpeech (Ren et al., 2019), 6 in this case. Similarly, there is a duration module, in this case, composed of a 3-layer convolution with an upsampling layer.

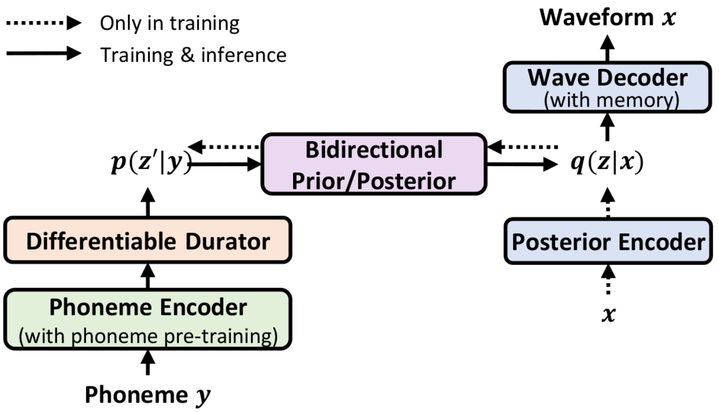


Figure 19: Architecture overview of NaturalSpeech from the original paper (Tan et al., 2022).

Summarizing, NaturalSpeech improves VITS by adding phoneme pre-training on a larger corpus, adds a differentiable durator operating at the frame level; it reduces the posterior, enhancing the prior using normalizing flow, and adds a memory-based VAE to improve the prior even further. Although the NaturalSpeech model is large, monolingual, and mono-speaker, due to its importance and relationship with the other models, we have considered it appropriate to give an overview of it. For more details, the reader can refer to the paper, which is rich in detail, to the code they have published[2], and to their website[3] with more information about their research.

2.3 Contributions

This research was initially proposed to train a multilingual Text-to-Speech model with support for Basque and Spanish. The idea was to use the YourTTS model, currently supporting English, Portuguese, and French, and see if we can add new languages to it without degrading its performance. As some languages like Basque do not have a comparable amount of resources, training it in a multilingual configuration may help the model learn it. As Spanish has similar phonological and phonetic linguistic rules, they may help each other get good results. In the same way, other minor languages from Spain, like Catalan and Galician, will be tested with the model. At the same time, having a multilingual model can alleviate the memory and resource usage of the model during inference since recent neuronal synthesis models, even though they give good results, have a considerable size. On top of that, a Speech-to-Speech Translation prototype has been designed using these synthesis models and other Deep Speech and Machine Translation models in a cascade set-up. The initial idea is to focus on the development model here and continue doing research on the other models later on.

[2] https://github.com/microsoft/NeuralSpeech

[3] https://speechresearch.github.io/naturalspeech/

2.4 Chapter Summary

In this chapter, we have seen a summary of models used for speech synthesis from the beginning of time to today's recent trends. The first machines capable of synthesizing audio began in the middle of the last century, starting with mechanical and more parametric methods that required an expert to configure and use. Subsequently, concatenative models arose, simpler but with limitations and requiring a lot of data to give good results. Afterward, statistical models were used, which combined flexibility with being intelligible (HMM), but still sounding unnatural. Lately, neural models have emerged and have been introduced, achieving a quality close to humans: which are formed by the classic feed-forward networks (FFN), convolutional neural networks (CNN), Recurrent Neural Networks (RNN), and the recent transformer. Finally, we have reviewed the state-of-the-art neural models that are working best, covering the different most used approaches, including flow-type (Glow-TTS), diffusion-type (Grad-TTS), and non-autoregressive models (VITS).

3 Methodology

In this section, a detailed description of all the processes involved in creating the Text-to-Speech synthesis models is provided. This includes the methods and tools used and created for preparing the datasets, configuring the models, performing the training steps, and the final evaluation of all the systems involved.

3.1 Project Setup

In order to do a proper project setup, a deep study, and analysis of the state-of-the-art models have been performed. We were searching for multilingual, multi-speaker models of medium size. All this process has been documented carefully and reviewed and guided by the tutors on a weekly basis. At the beginning of this work, the YourTTS model was indisputably the one that came closest to our needs.

The main goal of the project was to develop a model with support for at least three languages: English, Basque, and Spanish. For that, we needed suitable cleaned-up datasets with those three languages, hardware to train the models, and reviewers to evaluate the results generated by the models. In the following subsections, the process of these steps is described.

3.2 Data Preparation

As we were using a pre-trained model and decided to conserve the previous model language capabilities, we needed to both download the original datasets and add new datasets for the languages we were interested in adding. All the data have been pre-processed, including both the text and audio files. In the next subsections, the datasets used and the pre-processing methods are described.

3.2.1 Datasets

The English datasets re-used from the previous pieces of training are the VCTK and LibriTTS, with around 256 hours of audio data and 2,585 speakers. For Portuguese, the TTS-Portuguese (Casanova et al., 2022b) dataset has been reused with only one speaker and around 10 hours of speech. Similarly, for the French language, M-AILAB (Solak, 2017) has been included, with 170 hours of recordings and five speakers.

Additionally, other datasets have been included with the purpose of learning the new languages. For Basque, two datasets have been added: OpenSLR-76 (Kjartansson et al., 2020) and a dataset provided by the Aholab Signal Processing Laboratory research group called TTS-DB$_{EU}$ (Sainz et al., 2012), both of them reaching a total of 36 hours and 66 speakers. For Spanish, another two datasets provided by the Aholab have been included, TTS-DB$_{ES}$ (Sainz et al., 2012) and ELRA-TC, with a total of 30 hours of recording but only one new speaker. Something important to notice here is that all the speakers in TTS-DB$_{ES}$ are also in TTS-DB$_{EU}$ dataset, so when the Spanish language is added, only one new speaker is added to the whole dataset included in the training. Other Spanish datasets available online have been considered, but most of them mix Spanish accents from different countries without classifying or differentiating them. Being the Spanish language one of the most spoken languages in the world, it has many dialects and variations, and we thought mixing all these variants together could considerably increase language complexity and hinder the learning process. Therefore, many datasets have been discarded so as not to confuse the model. Without having ruled out including the entire range of dialects in a more than possible future work, we decided to progress with the Castillian variation of the Spanish language for now. For Galician and Catalan, OpenSLR-69 and OpenSLR-77 (Kjartansson et al., 2020) datasets have been included, having around 12 hours of recordings and 80 speakers. In Table 2, there are the details of each of the datasets summarized.

Dataset	Language	Recordings	Speakers	Female	Male	Time	Size
VCTK	English	44,453	110	63	47	6:12:53	3.6 GB
LibriTTS	English	155,471	1,620	889	731	249:52:01	29 GB
TTS-Portuguese	Portuguese	3,625	1	0	1	9:38:45	1.1 GB
M-AILAB	French	90,321	5	2	3	173:46:04	19 GB
TTS-DB$_{EU}$	Basque	20,698	14	7	7	26:57:27	6.4 GB
OpenSLR-76	Basque	7,136	52	29	23	9:01:56	2.0 GB
TTS-DB$_{ES}$	Spanish	16,158	5	3	2	19:05:28	4.5 GB
ELRA-TC	Spanish	5,432	1	1	0	9:48:13	2.2 GB
OpenSLR-69	Catalan	4,240	36	20	16	4:55:23	1.1 GB
OpenSLR-77	Galician	5,587	44	34	10	6:45:47	1.5 GB

Table 2: Datasets used for training the models. The size is after extracting the contents and having the recordings in wav format.

3.2.2 Text Pre-Processing

Text normalization is the process of converting an input text with a previously unknown form to a single canonical form that guarantees its consistency before using it for other purposes. It is a common process recommended before using it for speech synthesis models, as it helps them focus on producing speech rather than internal text transformations. Usually, this process includes converting numbers and dates to text, expanding acronyms and abbreviations, transformations to facilitate the pronunciation of foreign words, and the like.

As it is expected, this process may differ from language to language. And as we are dealing with different languages here, a python library called *normalize-text*[4] has been created as a wrapper for the different tools used depending on the language. This tool currently supports the following languages: Basque, Galician, Catalan, Arabic, Czech, German, English, Spanish, Farsi/Persian, French, Italian, Luxembourgish, Dutch, Russian,

[4] https://gitlab.com/xzuazo/normalize-text

Swedish, and Swahili. To be more specific, the tool uses an AhoTTS module for Basque (Hernaez et al., 2001); it also uses *Cotovia* for Galician (Rodríguez Banga et al., 2012), *FestCat* for Catalan (Bonafonte et al., 2009), and *gruut* (Hansen et al., 2022) for the rest of the languages. In the Examples below, sentence transformations can be seen as the library processes them. Example 3.1 uses Aholab's AhoTTS module to normalize a sentence in Basque. Example 3.2 uses Gruut for the Spanish language. As Spanish is supported by other underlying libraries like AhoTTS or Cotovia, this can be changed if required. We used Gruut for Spanish after some testing because it was the least aggressive normalization. The last Example 3.3 uses the FestCat Festival module to normalize Catalan using a short script in Scheme.

Example 3.1
Original text in Basque: Kaixo, gaur 2022/03/07 da.
Normalized text: Kaixo, gaur bi mila eta hogeita biko martxoaren zazpia da.

Example 3.2
Original text in Spanish: Hola, hoy es 3/7/2022.
Normalized text: Hola, hoy es tres de julio de dos mil veintidós.

Example 3.3
Original text in Catalan: Hola, avui és 3/7/2022.
Normalized text: Hola avui és tres de juliol del dos mil vint-i-dos.

In addition to the text normalization library mentioned above, the texts in Basque have been manually reviewed, and some of the subjective changes have been widely discussed with the tutors. For example, the OpenSLR-76 dataset includes many anglicisms, many more than are normally used in common speech. Because of this, many foreign names and words have been altered to be closer to the actual pronunciation based on the phonology rules of the language. To find these foreign words, a small tool has been developed that uses the fastText library (Mikolov et al., 2018) to predict the language of each word in each sentence. This library uses a set of publicly available pre-trained models for efficient text classification and representation learning that can be used for language detection, among other tasks. The foreign words found by the library have been manually reviewed and changed if the pronunciation in the recorded audio differed a lot from the text transcribed. A similar process has been carried out in the Spanish datasets. The rest of the datasets, however, have only been pre-processed by the automatic normalization tool.

3.2.3 Audio Pre-Processing

As we are using varied datasets from different sources, the formats of the files vary considerably. For training our Text-to-Speech model, all the audios have been normalized in multiple steps: format and volume homogenization, recording denoising, and silence trimming. As for the format, all the audios have been transformed to pulse code modulation (PCM) with a mono channel setup and a sample rate of 16 kHz with a bit-depth of 16 bits. For this, the *SoX* tool has been used. Afterward, silences have been removed with a script using the *webrtcvad*[5] python bindings for the Google WebRTC (Sredojev et al., 2015) Voice Activity Detection (VAD) module. Then, an additional denoising step has been performed using the causal speech enhancement model from Facebook (Defossez et al., 2020). Finally, the volume has been normalized to -27dB using the Root Mean Square-based (RMS) normalization using *ffmpeg-normalize*[6].

3.3 Text-to-Speech Training

For the training, an internal fork of the official Coqui-TTS (Eren and The Coqui TTS Team, 2021) has been used. The code has been slightly modified to fit our experiment setup and to train on multiple steps following an approach similar to the one used for the original YourTTS models. There are a total of four experiments, and each experiment is split into two pieces of training.

The first training is focused on learning the new language without the speaker cosine similarity loss, using Equation 4 for the loss. As languages may differ a lot, this step will take longer for some languages than for others, requiring more steps. Following the previous training of languages with this model, a total of 140k steps have been performed. In Equation 14, we can see how to calculate the number of epochs for each experiment. In our case, a batch size of 48 was used on a single GPU, and N refers to the number of instances of utterances or recordings, which varies in each experiment.

$$epochs = \frac{steps \cdot batch\_size \cdot gpus}{N} \tag{14}$$

The second piece of training is to learn to imitate the speakers, focused on taking the speaker cosine similarity loss into account without completely disabling the previous losses, using Equation 13. This step was shorter, taking only around 50k steps.

Both pieces of training were performed on an NVIDIA Titan RTX with 24 GB of memory, 576 tensor cores, and 4608 Cuda cores. Each of the experiments took around one week of training to complete, including both of the training pieces. From that full training process, the epoch with the best validation score was chosen for the final model.

3.4 Evaluation Method

Multiple evaluation methods have been used to test the quality of the speech synthesized from the different models trained. All of them will be evaluated using the same methods. Both automatic and manual evaluation methods will be used to test the quality of the generated audio and the speaker's similarity.

In this research, the Mean Opinion Score (MOS) will be used to evaluate the generated speech by having real human evaluators participate in listening and scoring the sentences through the Internet. MOS is a subjective measure to evaluate signal processing methods that can be used to approximate human perception by small studies.

[5] https://github.com/wiseman/py-webrtcvad

[6] https://github.com/slhck/ffmpeg-normalize

To test the abilities of the model to generate speech for specific speakers, audios from the ground truth will be compared with generated audios for the same speaker.

3.4.1 Automatic Evaluation

Automatic evaluation methods will use different existing models to get a score from the ground truth and generated audio files, and we will compare them. There exist multiple models that generate a score that tries to approximate the subjective MOS performed by real humans. VoiceMOS (Huang et al., 2022) is a challenge to find an automatic way to predict the mean opinion score where different research teams present their solutions and compete with each other. In the 2022 edition of the competition, they used different metrics to evaluate the system for each of the participants: system-level and utterance-level mean squared error (MSE), Linear Correlation Coefficient (LCC), Spearman Rank Correlation Coefficient (SRCC), and Kendall Tau Rank Correlation (KTAU). We decided to use two models from that competition: one the baseline SSL-MOS model (Cooper et al., 2021), and one of the models from the participants, UTMOS-22 (Saeki et al., 2022) that has the highest score on several metrics for both the primary track (test data from the same datasets as training data) and out-of-domain track (test data from datasets not seen in training). Additionally, the audios will be tested in the NISQA-TTS (v1.0) too (Mittag et al., 2021), an independent model to test synthesized speech quality from the naturalness point of view.

Besides, to check the speaker similarity between the ground truth audios and the generated speech, Speaker Encoder Cosine Similarity (SECS) (Casanova et al., 2021) will be checked on multiple speakers from both the Spanish and Basque datasets. This algorithm checks the cosine similarity between generated speaker embeddings. The resulting score goes from -1 to 1, with bigger values indicating stronger similarity. Based on YourTTS work (Casanova et al., 2022b), the speaker embedding will be generated using the Resemblyzer speaker encoder (Wan et al., 2017).

There is an additional metric used internally to debug some problems we found on some of the models, which we called *Regex-Error-Rate* (RER). We found some of the models have problems pronouncing some characters correctly at the end of the sentence, so Speech-to-Text models have been used to evaluate the correctness. The most problematic phones were the voiced alveolar tap [ɾ] and the voiced alveolar trill [r] corresponding to the "r" character pronunciation. For that, we first generated a Text-to-Text speech model merging in cascade our Text-to-Speech (the model to evaluate) and the Speech-to-Text model. This model generates speech from text and converts it back to text. The Speech-to-Text models used for this are DeepSpeech models, some of them available online, others trained in-house (see Chapter 5.3 for details). The full alphabet for Basque was tested using the Text-to-Text speech model, investigating whether other characters may have a similar problem: the "j" character had a similar problem, but that character is not very common in Basque at the end. Checking the rest of the alphabet, the other characters had no problem. So we focused on the *R-Error-Rate* metric, sampling sentences ending in "r" character using the `".*r[.]?$"` regular expression and passing them through our Text-to-Text model. We synthesized a total of 1000 sentences for both Basque and Spanish. Then we checked if the output transcription still passed the regular expression. This metric can be used to evaluate sentences ending in other characters or meeting other different regular expression criteria. Still, we will focus mainly on the "r"-ending sentences here.

3.4.2 Human Evaluation

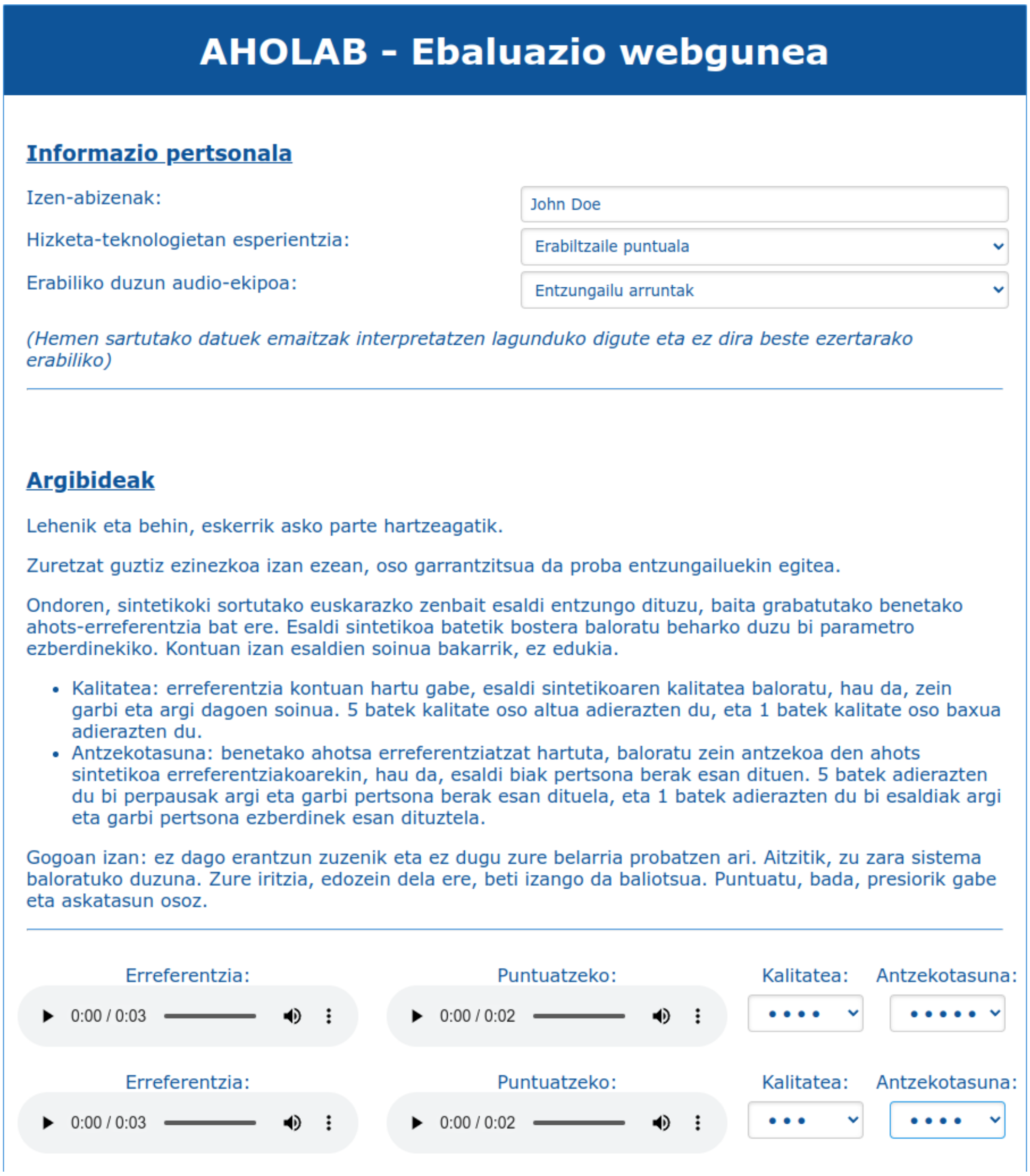
Human MOS scores have also been gathered using an online Web MOS Evaluation Interface provided by the Aholab research team [7]. In Figure 20, we can see how the final interface looks like for the evaluators, for the Basque language in this case. After writing their personal information, the users need to listen to the reference audio (*Erreferentzia*) and the audio to evaluate (*Puntuatzeko*) and give a score from 1 to 5 for the naturalness (*Kalitatea*) and speaker similarity (*Antzekotasuna*). The interface has been updated to improve cross-browser and mobile device support. Additionally, multi-language support has also been added to translate the explanations to the language being tested by the evaluator.

With the help of this tool, we calculated both the MOS for speech naturalness and the Sim-MOS for speaker similarity. We evaluated using speakers from the training dataset and external speakers in separate tests to know the models' performance in real-life scenarios with unknown voices. The external speakers were provided by the ZureTTS voices bank (Erro et al., 2014, 2015). During the evaluation, we used 16 native Basque speaker contributors and 24 native Spanish contributors. For each of the models, we synthesized 50 sentences for both Spanish and Basque. The Spanish sentences have been sampled from the phonetically balanced Sharvard corpus (Aubanel et al., 2014). For the Basque sentences, the 50 sentences used for the Spanish sentences have been translated by hand. Then the synthesized sentences were evaluated through the MOS evaluation interface by the evaluator's team of the Aholab research group.

3.5 Prototype Development

The online prototype is a web interface around the Text-to-Speech and Speech-to-Text models used here to generate Speech-to-Speech Translation (S2ST). All the Text-to-Speech models were trained during the research here. Some of the Speech-to-Text models were trained during our research here; others were downloaded online. Moreover, it includes Machine-Translation models on inference configuration. All three different modules can be tested on different sections. Additionally, thanks to this modular architecture, a Speech-to-Speech mode can be tested that uses the trained models underneath. With this, a speech-to-speech language translation can be performed, both conserving the speaker's voice or also selecting between a list of speakers from the training dataset. For more details about the development of this prototype, check Chapter 5.

[7] https://aholab.ehu.eus/users/xzuazo/mos/eu/



AHOLAB - Ebaluazio webgunea

Informazio pertsonala

Izen-abizenak: John Doe

Hizketa-teknologietan esperientzia: Erabiltzaile puntuala

Erabiliko duzun audio-ekipoa: Entzungailu arruntak

(Hemen sartutako datuek emaitzak interpretatzen lagunduko digute eta ez dira beste ezertarako erabiliko)

Argibideak

Lehenik eta behin, eskerrik asko parte hartzeagatik.

Zuretzat guztiz ezinezkoa izan ezean, oso garrantzitsua da proba entzungailuekin egitea.

Ondoren, sintetikoki sortutako euskarazko zenbait esaldi entzungo dituzu, baita grabatutako benetako ahots-erreferentzia bat ere. Esaldi sintetikoa batetik bostera baloratu beharko duzu bi parametro ezberdinekiko. Kontuan izan esaldien soinua bakarrik, ez edukia.

- Kalitatea: erreferentzia kontuan hartu gabe, esaldi sintetikoaren kalitatea baloratu, hau da, zein garbi eta argi dagoen soinua. 5 batek kalitate oso altua adierazten du, eta 1 batek kalitate oso baxua adierazten du.
- Antzekotasuna: benetako ahotsa erreferentziatzat hartuta, baloratu zein antzekoa den ahots sintetikoa erreferentziakoarekin, hau da, esaldi biak pertsona berak esan dituen. 5 batek adierazten du bi perpausak argi eta garbi pertsona berak esan dituela, eta 1 batek adierazten du bi esaldiak argi eta garbi pertsona ezberdinek esan dituztela.

Gogoan izan: ez dago erantzun zuzenik eta ez dugu zure belarria probatzen ari. Aitzitik, zu zara sistema baloratuko duzuna. Zure iritzia, edozein dela ere, beti izango da baliotsua. Puntuatu, bada, presiorik gabe eta askatasun osoz.

Erreferentzia:	Puntuatzeko:	Kalitatea:	Antzekotasuna:
0:00 / 0:03	0:00 / 0:02	••••	•••••
0:00 / 0:03	0:00 / 0:02	•••	••••

Figure 20: Web MOS Evaluation Interface for the Basque language.

3.6 Chapter Summary

In this chapter, the methodology followed to complete this research has been described from beginning to end. Firstly, the dataset selection and preparation, including text and audio pre-processing tasks. Then the model preparation and configuration, with training details, required software and hardware, and approximate time required. Finally, the different evaluation methods chosen to assess the quality of the final Text-to-Speech models have been presented. In the next chapter, the specific setup and results of each of the experiments will be presented.

4 Findings

In this chapter, the training progress of each experiment is shown, together with the evaluation results and some findings. We have devised a total of four main experiments to train a multilingual Text-to-Speech model incrementally, adding new languages in each experiment and checking if the model continues to progress. The complete process consists of training for Basque in the first experiment, Spanish in the third experiment, and Catalan and Galician in the third experiment; the fourth and latest experiment is just an extension of the learning process without new information. In the last section, some additional experiments will be described that were finally discarded because they did not reach the expected results. Still, we consider them relevant because they explain the decisions taken in the other experiments.

All the experiments use the same YourTTS model (Casanova et al., 2022b), and most of them have a very similar setup; most of the changes are related to the datasets, cleaning the examples properly and adjusting the steps. At the same time, each of the experiments is split into two pieces of training: a first longer training step focused on learning the new language of around 140k steps, and a second shorter training step focused on learning the new speakers of around 50k steps. The latter is called Speaker Consistency Loss training (SCL) and is where the SCL loss is activated (see Equation 12 for more information).

As we progress in each experiment, the new model will be evaluated and compared with the previous experiment. Evaluation metrics are both audio naturalness and speaker similarity related. Automatic metrics are generated by computer models and try to approximate real human perception of the generated speech. In addition, the ”r”-ending sentences Error Rate also will be evaluated using a Speech-to-Text model. In total, we provide three automatic evaluation metrics for audio naturalness (SSL-MOS, UTMOS-22, and NSIQA) to provide three independent points of view, an automatic evaluation metric for speaker similarity (SECS), and an R-Error-Rate score. Human metrics have been evaluated by the Aholab group evaluation team using a web evaluation interface (see Section 3.4 for more details), and they provide both naturalness MOS and speaker similarity MOS (Sim-MOS). Values given for the scores will be the mean and the 95% confidence interval when available.

4.1 Experiment 1: Basque Language

In this experiment, we will part from a multilingual model provided by the Coqui-TTS team that already supports three languages: English, Portuguese, and French. The idea is to add Basque support to this model following the training process described in the YourTTS original paper (Casanova et al., 2022b). This will be done in a way that does not forget previously learned languages. In other words, the Text-to-Speech model produced here will support four languages in total: English, French, Portuguese, and Basque.

4.1.1 Experimental Setup

For this experiment, two Basque audio datasets with transcriptions have been added: TTS-DB (provided by Aholab) and OpenSLR-76 (Kjartansson et al., 2020). These datasets contain a total of 36 hours of audio recordings and 66 speakers. During the pre-processing of both datasets, metadata files have been generated using BRSpeech-based format (Casanova et al., 2022a,b) for each of the datasets. The format consists of a tab separate value file with the audio file relative path, the pre-processed text, the post-processed text, and the speaker. In the original BRSpeech format, the second field is the file size in bytes, which is a format frequently used for STT tasks, but here we decided to use the pre-processed text to aid in the laborious text normalization task. Additionally, in order for the model not to forget the previous languages and speakers, the following datasets have been prepared and pre-processed as described in YourTTS paper (Casanova et al., 2022b): VCTK (Christophe Veaux and MacDonald, 2017) and LibriTTS (Zen et al., 2019) for English, TTS-Portuguese (Casanova et al., 2022a) for Portuguese and M-AILABS (Solak, 2017) for French. All these datasets have been pre-processed in the exact same way, using the developed `normalize-text` tool for the text and an audio normalizing process of three steps: silence trimming, audio denoising and volume normalization (see Section 3.2 for more details). Including all the datasets, this model will be trained in around 475 hours of audio recordings and 1310 speakers.

To make the training faster, we used transfer learning from a pre-trained YourTTS model. This model was already trained for English, Portuguese, and French using the same datasets. For the first step of this experiment, 262,121 instances of recordings will be used in total for training with a sequence length between 30 and 250 characters with an average sequence length of 91 characters. Following Equation 14, we will do a first training step to learn the Basque language of 26 epochs (5460 steps/epoch). In this step, the batch size used has been reduced to 48 due to hardware constraints. The loss used here will be the one used for the original VITS model (Kim et al., 2021) (see Equation 4 for more details).

For the second step focused on learning new speakers (SCL), 179,426 instances will be used of length between 60 and 270 characters, with the same sentence average length as in the previous training. Following the same equation, we will train the SCL model for 11 epochs (4600 steps/epoch). The batch size used for this second step is reduced to 39, and the α value of SCL loss has been set to 9 as in the YourTTS paper (Casanova et al., 2022b). Therefore, the loss used for this step will be YourTTS model loss, including the speaker cosine similarity loss (L_{SCL}, see Equation 13 for more details).

4.1.2 Evaluation and Result

The duration of the full experiment training took around 4 days for the first step and the second step around a day and a half, making a total of 5.5 days to complete the whole experiment.

In Figure 21, we can see the full training plots of Experiment 1, including all the losses for train and development splits with its internal name in both official VITS [8] and Coqui-TTS [9] implementations. *Loss disc* refers to the HiFi-GAN Discrimination Loss ($L_{adv}(D)$) presented in Equation 9: this loss is supposed to increase, as the discriminator is supposed to have more difficulties in differentiating between real and generated examples as the training progresses. *Loss duration* is the Duration Loss ($L_{dur}(D)$) presented in Equation 8, and as we see, it has difficulties improving, probably because it already knows common lengths of phonemes from previous pieces of training. *Loss feat* shows the Feature Loss ($L_{fm}(G)$) of the HiFi-GAN presented in Equation 11, it measures the differences in features between the discriminator and the generator on each layer of the GAN: here we can see how the loss decreases considerably, particularly in the first 50k steps. *Loss gen* refers to the Generation Loss ($L_{adv}(G)$) of the GAN presented in Equation 10. *Loss kl* is the Kullback–Leibler (KL) divergence loss (L_{kl}) explained in Equation 6 to compare entropy between the input linear spectrogram and characters with the estimated alignment created by the flow: we can see how the model finds some difficulties to decrease this loss, maybe due to the statistical complexity increase as we add new languages. The last loss is *Loss mel* and refers to the mel-spectrogram Reconstruction Loss (L_{recon}) presented in Equation 5: this loss has a slight but continuous descent in the development loss.

Finally, we have the *Loss 0* and *Loss 1* that gather together the losses that need to decrease for the former and the losses that need to increase for the latter. The *Loss 0* is the same as the combination of the VAE and GAN training final loss we presented in Equation 4 as L_{vae}: summing up *Loss duration*, *Loss feat*, *Loss gen*, *Loss kl*, and *Loss mel*. The *Loss 1* is just the same as the Discriminator Loss or the *Loss disc* recently explained. Both of these losses have the expected trend when putting together all the losses just explained.

In Figure 22, the training plots of the Experiment 1 SCL part are presented. During this training, the model is focused on training the new speakers more than the language. As it is shown, there is a new loss named *Loss spk encoder* that was included in the VITS model implemented by YourTTS paper (Casanova et al., 2022b). This new loss refers to the Speaker Consistency Loss (L_{SCL}) introduced in Equation 12 and added to the *Loss 0* as previously shown in Equation 13. Checking the development scores, it can be seen that the Speaker Consistency Loss is the main making improvement; the other losses do not seem to change much.

In Table 3, we can see the naturalness scores of both pieces of training in this experiment. The scores have been generated by the VoiceMOS 2022 baseline SSL-MOS model. The ground truth scores correspond to the scores of real recordings from the dataset, and YourTTS are the scores of the pre-trained model before adding the Basque language through this experiment. The underlined scores mean that they outperformed ground truth scores, and the best scores are marked in bold.

Checking the ground truth scores, we can see that the quality for both VCTK and LibriTTS is quite lower than the English datasets, even below a score of 4 in MOS. This is because the MOS on MLS-PT and TTS-DB$_{EU}$ should be considered out-of-domain, as

[8] https://github.com/jaywalnut310/vits

[9] https://github.com/coqui-ai/TTS/tree/dev/TTS/tts/models

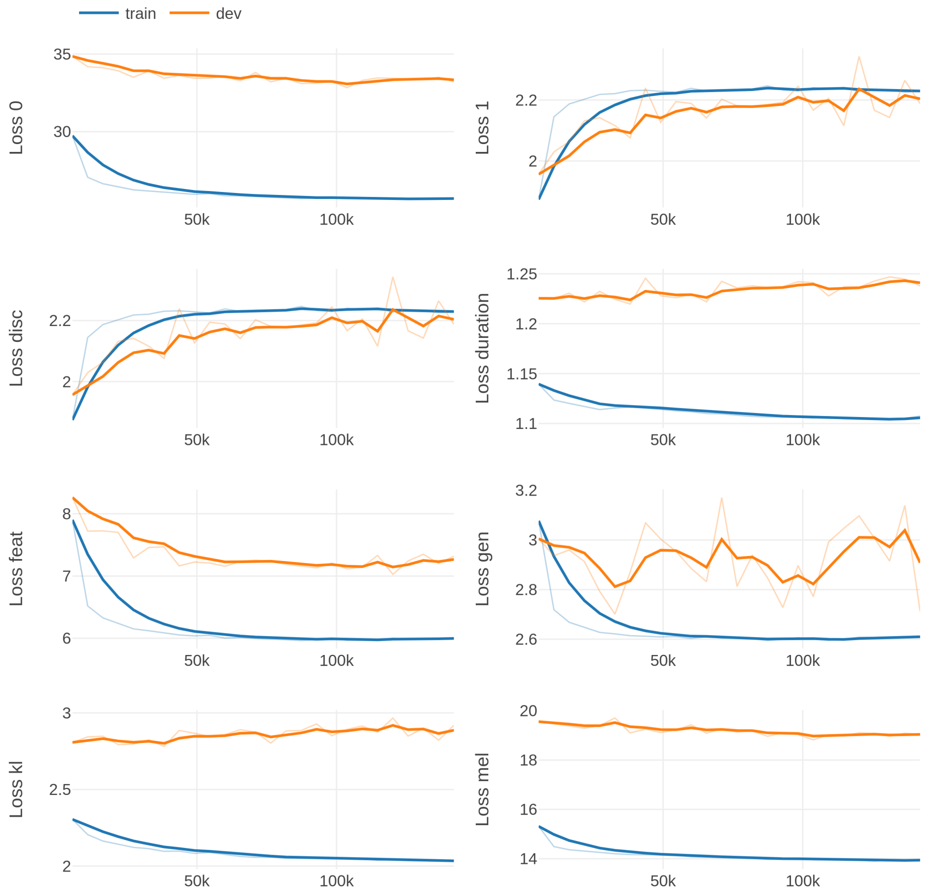


Figure 21: Experiment 1 training plot, including all the losses for train and development splits with its internal name in both official VITS and Coqui-TTS implementations.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**
Ground truth	*4.08±0.09*	*4.42±0.04*	*2.46±0.10*	*3.19±0.11*
YourTTS	4.00±0.11	4.08±0.10	2.45±0.09	
Exp.1	**4.23±0.07**	**4.18±0.10**	**3.42±0.14**	3.04±0.09
Exp.1+SCL	4.20±0.09	4.16±0.11	2.69±0.09	**3.10±0.10**

Table 3: Mean Opinion Score (MOS) approximation by VoiceMOS 2022 baseline SSL-MOS model after 1st experiment.

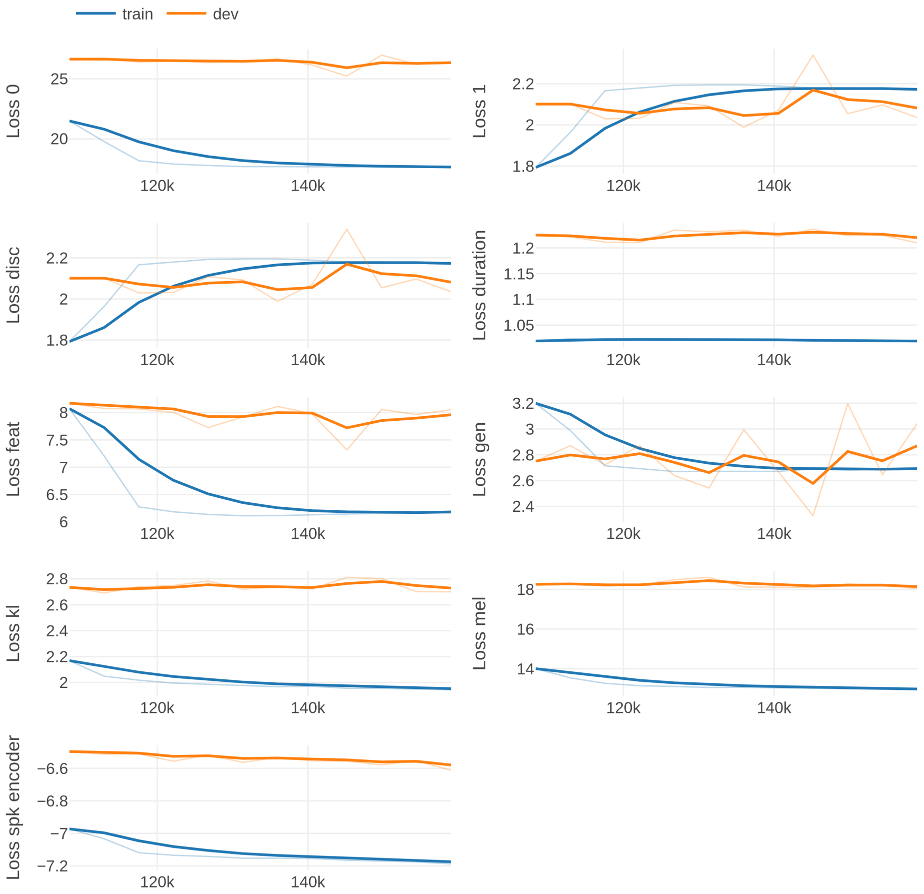


Figure 22: Experiment 1 SCL training plot, including all the losses for train and development splits with its internal name.

SSL-MOS has not been trained on those languages. Checking the experiment results in English datasets, we obtained better results than ground truth for the VCTK dataset and not-so-good results for the LibriTTS dataset. However, still, all the scores outperformed the pre-trained YourTTS model. Moreover, we can see it also got better results than ground truth on the MLS-PT dataset. This may have some underlying reason, as the Portuguese dataset was recorded by a single speaker with non-professional equipment, being its quality not comparable to the other datasets. Training the model on more datasets of better quality may have affected the synthesis of the Portuguese language by improving it, as the model may have developed some cross-language naturalness learning ability. As for the Basque Language in TTS-DB, the scores are lower than ground truth, and all the scores are pretty low, probably for being an out-of-domain evaluation. Comparing it with the previous YourTTS model, scores seem to have improved overall. In general, it seems that the second training step (SCL) obtained worse scores; this may have some sense, as in this step, the model is more focused on learning the new speakers than on sound naturalness.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**
Ground truth	*4.03±0.05*	*4.31±0.03*	*2.79±0.08*	*3.59±0.10*
YourTTS	3.63±0.08	3.70±0.09	2.62±0.10	
Exp.1	**3.80±0.07**	**3.79±0.07**	**3.35±0.10**	3.13±0.08
Exp.1+SCL	3.79±0.08	3.78±0.09	2.78±0.10	**3.18±0.09**

Table 4: Mean Opinion Score (MOS) approximation by UTMOS-22 model after 1st experiment.

In Table 4, we can see the naturalness scores of both pieces of training generated by the VoiceMOS 2022 participant UTMOS-22 model. As with the SSL-MOS model, we see that Experiment 1 got better scores than the YourTTS model overall. The Portuguese-generated speech in MLS-PT also gets much better. Still, the second step of the training has worse results while focusing on training the speakers. An important difference to point out compared with previous SSL-MOS results is that English scores using VCTK and LibriTTS are much worse than the ground truth, not even reaching a score of 4.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**
Ground truth	*3.75±0.16*	*3.98±0.14*	*3.31±0.18*	*4.11±0.12*
YourTTS	3.37±0.10	**3.53±0.12**	**3.19±0.16**	
Exp.1	**3.43±0.11**	3.47±0.13	3.13±0.13	**3.66±0.15**
Exp.1+SCL	3.42±0.10	3.43±0.13	3.12±0.13	3.65±0.15

Table 5: Mean Opinion Score (MOS) naturalness approximation by NISQA v1.0 after 1st experiment.

In Table 5, there are other naturalness evaluation results, this time by the NISQA v1.0 model. This model gives very low scores, too, even for the in-domain English dataset on the ground truth. There is no clear winner in these scores: between the original YourTTS and the Non-SCL Experiment 1, the scores are tight and within the confidence interval. Some quality degradation is appreciated in the SCL experiment scores, but it is still very small.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**
Ground truth	*0.824±0.018*	*0.932±0.008*	*0.901±0.015*	*0.900±0.007*
YourTTS	0.845±0.009	0.858±0.011	0.803±0.010	
Exp.1	**0.849±0.009**	0.862±0.014	0.813±0.012	0.873±0.023
Exp.1+SCL	0.845±0.010	**0.868±0.013**	**0.817±0.012**	**0.879±0.027**

Table 6: Speaker Encoder Cosine Similarity (SECS) after 1st experiment.

In Table 6, speaker cosine similarity metrics are presented to see how the models are learning the speakers for each of the datasets. The TTS-DB$_{EU}$ is the only new dataset, so the others were already known by the pre-trained model. The scores show that, except for the VCTK English dataset, the speaker similarity improved slightly with the SCL experiment. Still, the difference is very low and within the confidence interval, so probably not much appreciable. Indeed, for the VCTK dataset, the SCL scores improved the ground truth scores calculated from real recordings. Apart from that, the improvements over the previous YourTTS model are not big but appreciable.

Experiment	**CER**	**SpkM1**	**SpkF1**	**SpkM2**	**SpkF2**	**SpkM3**	**SpkF3**	**Mean**
Ground truth	19.9%							19.1%
Exp.1	20.8%	80.9%	94.5%	93.9%	99.4%	**84.7%**	95.8%	79.1%
Exp.1+SCL	19.6%	**80.0%**	**92.2%**	**90.5%**	**96.5%**	91.1%	**94.0%**	**76.0%**

Table 7: ”R” at the end problem evaluation on TTS-DB dataset with the Basque language. The ground truth row represents the Speech-to-Text overall performance in the dataset. The Mean represents the mean between all the speakers, not just the 6 included here.

Last but not least, in Table 7, the measure R-Error-Rate can be appreciated using an audio transcription model. In the table presented, the individual RER for 6 speakers for the TTS-DB dataset in Basque is shown, and the mean, which includes all the speaker's RER averaged. The Speech-to-Text model used has a Character Error Rate of 19.9%, corresponding to the score of the Basque STT Deep Speech model v0.1.4 (see section 5.3.1) without a Language Model transcribing 1000 randomly chosen sentences from the TTS-DB Basque dataset. The general CER of the model transcribing sentences from the TTS-DB dataset has been calculated too, following a similar approach of synthesizing 1000 random sentences with a random speaker. As we can see, the experiment CER remains near the ground truth CER, even reaching a lower value in the SCL experiment. On the other hand, the Speech-to-Text model has an R-Error-Rate of 19.1% in the TTS-DB dataset, calculated by checking real recording transcriptions in the same dataset. Since this last value is on par with the CER value, it can be considered that the ”r” is not a particularly difficult character for the STT model to recognize and can be used as a good measure for the ground truth. The language model of the Speech-to-Text model has been disabled to avoid statistical word corrections of the transcriptions and focus only on the audio phonetic characteristics for transcription. For the evaluation, 6 speakers have been selected that we have confirmed have this problem frequently, 3 female and 3 male. The mean values shown are averaged over all the speakers in the dataset, which may include speakers in which this problem is not so recurrent. Basically, this evaluates the correct pronunciation of the "r" at the end of the sentence using a Speech-to-Text model to check the sentences. As it can be appreciated, most of the time, the "r" is not correctly pronounced at the end of the sentences. The error seems to be more prominent with female speakers here, but all of them have very high error rates.

To check the real dimension of the problem of bad pronunciation of R at the end of the sentences, we listened to the problematic sentences previously. In some cases, a very soft "r" can be perceived, even if the Speech-to-Text model has not detected it; in other cases, it is entirely absent. The full alphabet has been checked, and this problem occurs mainly with this character and at the end of the sentences. Let us continue exploring this problem in the next pieces of training.

4.2 Experiment 2: Spanish Language

For this experiment, we do transfer learning from the SCL model obtained in Experiment 1 (see section 4.1) trained in English, Portuguese, French, and Basque, and we will add the Spanish language support. For that, Spanish recordings will be added without removing the previous datasets used in Experiment 1. With that approach, we expect the multilingual TTS model to learn to synthesize Spanish speech without forgetting the previous languages.

4.2.1 Experimental Setup

In this second experiment, two datasets have been added, as in Experiment 1; only in this case, the recordings are in Spanish. The datasets contain both recorded speech and their transcriptions. The datasets have been provided by the Aholab group and are the following: TTS-DB and ELRA-TC. TTS-DB is the same dataset used for Basque, but that also contains some Spanish recordings: from a total of 14 speakers in TTS-DB, 5 of them have recordings in Spanish too. This means that those speakers are not new and were also used for the Basque training in Experiment 1. The ELRA-TC dataset contains only one speaker. Therefore, only one new speaker will be added, reaching a total of 1311 speakers. The recording time added by this dataset is around 30 hours, so the amount of the total recording time increased to a value of 505 hours. These new datasets have also been pre-processed with `text-normalize` package for the transcriptions and the three-step audio normalizing process described in Experiment 1 as explained in section 4.1.

This experiment is also split into two training pieces: one to learn the new language and a second one to focus on learning the speakers. For the first part, a total of 280,689 recording instances have been used with an average sentence length of 91 characters; sentences shorter than 30 characters and longer than 250 characters have been discarded. Based on

Equation 14, the model will be trained for 24 epochs to learn Spanish (5847 steps/epoch). The batch size has not been changed from Experiment 1, which is still 48.

The SCL part of the training will take care of learning the new speaker. Even though there is only one new speaker, we still tried to train it for a similar length as in Experiment 1. With a minimum sequence length of 60 characters and a maximum sequence length of 270 characters, we got a total of 190,117 recordings. The SCL model has been trained for a total of 11 epochs (4874 steps/epoch). The batch size is still 39, as in the previous experiment, and the α value of SCL loss (L_{SCL}) has been set to 9 again (see Equation 12 for more details).

4.2.2 Evaluation and Result

The first part of the training took approximately 4 days to complete, and the second part 1 day, making a total of 5 days of training.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.08±0.09*	*4.42±0.04*	*2.46±0.10*	*3.19±0.11*	*3.16±0.16*
Exp.1+SCL	4.20±0.09	**4.16±0.11**	2.69±0.09	3.10±0.10	
Exp.2	4.18±0.10	4.11±0.13	2.73±0.08	**3.14±0.11**	2.99±0.17
Exp.2+SCL	**4.21±0.08**	4.12±0.09	**2.77±0.10**	3.09±0.13	**3.06±0.14**

Table 8: Mean Opinion Score (MOS) approximation by VoiceMOS 2022 baseline SSL-MOS model after 2nd experiment.

In Table 8, the naturalness scores by the VoiceMOS 2022 baseline SSL-MOS model are shown, compared with the previous experimental results. As before, the best results are shown in bold and underlined if they are better than the ground truth. On balance, this experiment seems to have better scores than the previous experiment. Just LibriTTS went a little worse; that is a complex dataset because it has recordings from many different speakers. Still, the difference between the scores is pretty small and inside the confidence interval. TTS-DB dataset scores got similar results for Basque and Spanish, which makes sense because they were recorded using the same hardware and software, and the quality must be similar. This may imply that recording quality affects the learning capacity of the model, looking that the scores and both TTS-DB datasets more or less match.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.03±0.05*	*4.31±0.03*	*2.79±0.08*	*3.59±0.10*	*3.62±0.15*
Exp.1+SCL	3.79±0.08	3.78±0.09	2.78±0.10	3.18±0.09	
Exp.2	3.76±0.08	**3.79±0.10**	2.91±0.09	**3.26±0.10**	3.13±0.18
Exp.2+SCL	**3.80±0.09**	3.74±0.07	**2.94±0.09**	3.20±0.11	**3.18±0.15**

Table 9: Mean Opinion Score (MOS) approximation by UTMOS-22 model after 2nd experiment.

In Table 9, there are other naturalness scores, this time provided by VoiceMOS 2022 UTMOS-22 model. Here also, the new experiment scores have improved slightly. Another aspect to notice is that MLS-PT scores are better than ground truth, as happened in the non-SCL results of Experiment 1. As before, the scores are very low, especially with the out-of-domain languages, not reaching a score of 4 even in the ground truth. This may mean that the scores of non-English languages are not so meaningful. In contrast to the previous training, this time, the SCL model seems to be behaving better than the non-SCL model.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*3.75±0.16*	*3.98±0.14*	*3.31±0.18*	*4.11±0.12*	*3.62±0.28*
Exp.1+SCL	3.42±0.10	3.43±0.13	3.12±0.13	3.65±0.15	
Exp.2	3.50±0.11	3.54±0.13	**3.31±0.14**	3.67±0.15	3.40±0.24
Exp.2+SCL	**3.60±0.11**	**3.58±0.13**	3.23±0.14	**3.72±0.14**	**3.45±0.21**

Table 10: Mean Opinion Score (MOS) naturalness approximation by NISQA v1.0 after 2nd experiment.

In table 10, we can see the naturalness scores returned by the NISQA v1.0 model in Experiment 2. This time the SCL part of Experiment 2 seems to have improved the results a little. In general, the scores of this NISQA model seem to be too low, not reaching a score of 4 even for the ground truth recordings, except for TTS-DB$_{EU}$. In fact, the difference in the scores obtained by NISQA in TTS-DB$_{EU}$ and TTS-DB$_{ES}$ datasets are somewhat unexpected since both datasets were recorded with the same speakers using the same equipment.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*0.824±0.018*	*0.932±0.008*	*0.901±0.015*	*0.900±0.007*	*0.884±0.014*
Exp.1+SCL	0.845±0.010	**0.868±0.013**	**0.817±0.012**	0.879±0.027	
Exp.2	0.843±0.010	0.860±0.012	0.813±0.013	**0.882±0.021**	0.871±0.028
Exp.2+SCL	**0.847±0.011**	0.861±0.012	0.811±0.012	0.877±0.029	**0.871±0.025**

Table 11: Speaker Encoder Cosine Similarity (SECS) after 2nd experiment.

The speaker similarity automatic score is shown in Table 11. The differences between the experiments are very low, and there is no clear winner here. The reason may be that this dataset only contains one new speaker, so there is not much new to learn. The other speakers from TT-DB$_{ES}$ were already in TT-DB$_{EU}$, so not much improvement can be made by the model. Additionally, the difference in scores falls between the confidence interval, so no clear conclusion can be drawn.

As for the previously mentioned R-Error-Rate, there is a clear improvement. In Table 12, we can see the percentage of error detected by the Speech-to-Text model. The mean error rate went down from 76% to 56% by adding the Spanish dataset and training for longer. Even though there is a clear decrease in the R-Error-Rate, the scores are still very

Experiment	CER	SpkM1	SpkF1	SpkM2	SpkF2	SpkM3	SpkF3	Mean
Ground truth	19.9%							19.1%
Exp.1+SCL	19.6%	80.0%	92.2%	90.5%	96.5%	91.1%	94.0%	76.0%
Exp.2	20.0%	63.5%	78.2%	80.8%	99.4%	**41.8%**	88.3%	61.1%
Exp.2+SCL	19.6%	**59.9%**	**64.5%**	**76.8%**	**95.1%**	50.9%	**76.9%**	**55.9%**

Table 12: "R" at the end problem evaluation on TTS-DB dataset with the Basque language after 2nd experiment.

high, all of them above 50% and far from the mean transcription error rate of 19%. The CER of the new models still remains around the CER of the STT model on real data, and again the SCL model has even improved the ground truth CER scores.

As a summary of the results of this experiment, not only did we add a new language to it, but also the overall scores seem to be improved for the previously known languages as well. This may be due to the addition of more data to train, the longer training time, or both.

4.3 Experiment 3: Catalan and Galician Language

In this experiment, something new will be tried: learning two new languages at the same time. Starting from our previous model trained on English, Portuguese, French, Basque, and Spanish, 5 languages in total, using transfer learning, we will try to teach the model two new languages: Catalan and Galician. These languages have some phones and transcription rules in common with previous languages, so we expect the model not to have great difficulties. To do this experiment, again, the previous datasets will be reused, adding to them two new datasets: one for Catalan and another for Galician.

4.3.1 Experimental Setup

The two datasets used for this experiment will be OpenSLR-69 for Catalan and OpenSLR-77 for Galician (Kjartansson et al., 2020). The OpenSLR-69 contains 36 speakers, and OpenSLR-77 47 new speakers, in total adding 5 and 7 hours of recordings to the complete dataset, respectively. These new dataset transcriptions have been pre-processed using `normalize-text`, which uses *Cotovia* and *FestCat* in the background for text normalization, the former for Galician and the latter for Catalan. Additionally, all the audios have been normalized as with the previous datasets: silence trimming, denoising, and normalizing volume.

The experiment will follow the same approach as before, splitting the training into two parts. The first will be longer and focus on learning the 2 new languages; the second part will be to learn the speakers. For the first part, we got a total of 289,203 instances after removing sentences shorter than 30 characters and longer than 250 characters. Following Equation 14, the model will be trained for 29 epochs (4874 steps/epoch). The batch size has been maintained at 48.

The second part of the experiment consists in focusing on the speakers. There are many new speakers, yet we decided to continue training for the same steps as before. Removing sentences longer than 270 characters and shorter than 60 characters, a total of 194,637 sentences have been used. The average sentence length for both parts is around 89 characters per sentence. Based on Equation 14, the model needs to be trained for 11 epochs (4990 steps/epoch). The α value of the L_{SCL} loss has been maintained at 9.

4.3.2 Evaluation and Result

The training took around 4 days for the first part and 2 days for the second part, making a total of 6 days: a little longer than before.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.08±0.09*	*4.42±0.04*	*2.46±0.10*	*3.19±0.11*	*3.16±0.16*
Exp.2+SCL	**4.21±0.08**	**4.12±0.09**	2.77±0.10	3.09±0.13	3.06±0.14
Exp.3	4.07±0.08	4.04±0.10	2.74±0.09	3.03±0.12	3.02±0.12
Exp.3+SCL	4.19±0.09	4.06±0.10	**2.78±0.09**	**3.14±0.12**	**3.16±0.16**

Table 13: Mean Opinion Score (MOS) approximation by VoiceMOS 2022 baseline SSL-MOS model after 3rd experiment.

In Table 13, we can see the estimated MOS by the SSL-MOS model. In this case, the situation is a little different from the previous experiments. The English language speech quality decreased slightly. However, the Basque and Spanish language's speech quality continued to improve by a notch. Another fact to highlight is that the Spanish speech synthesis reached the quality of the ground truth. This may be because there is a high similarity between the Catalan, Galician, and Spanish languages phonetically. Comparing both models from the experiment, the SCL seems to be a little better with out-of-domain languages, getting little worse scores for English.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.03±0.05*	*4.31±0.03*	*2.79±0.08*	*3.59±0.10*	*3.62±0.15*
Exp.2+SCL	3.80±0.09	**3.74±0.07**	2.94±0.09	3.20±0.11	3.18±0.15
Exp.3	3.70±0.06	3.71±0.09	**3.02±0.09**	3.18±0.10	3.13±.013
Exp.3+SCL	**3.82±0.08**	3.72±0.08	2.98±0.10	**3.29±0.10**	**3.29±0.15**

Table 14: Mean Opinion Score (MOS) approximation by UTMOS-22 model after 3rd experiment.

The speech quality scores by the UTMOS-22 model can be appreciated in Table 14. Even though all the scores are very low, never reaching 4, the overall tendency is for improvement. There are still some difficulties in improving the synthesis of English using speakers from LibriTTS. As this dataset is big and with many different speakers, it is frequently the most challenging; there may not exist much room for improvement. The

Portuguese synthesis continues to be better than ground truth, probably due to out-of-domain difficulties. As with the previous scores, here, the SCL version of the experiment seems to be a little better, but still, the difference is within the confidence interval.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*3.75±0.16*	*3.98±0.14*	*3.31±0.18*	*4.11±0.12*	*3.62±0.28*
Exp.2+SCL	**3.60±0.11**	3.58±0.13	3.23±0.14	3.72±0.14	**3.45±0.21**
Exp.3	3.40±0.12	3.40±0.13	3.27±0.17	3.49±0.15	3.26±0.20
Exp.3+SCL	3.50±0.12	**3.63±0.13**	**3.39±0.15**	**3.74±0.16**	3.44±0.23

Table 15: Mean Opinion Score (MOS) naturalness approximation by NISQA v1.0 after 3rd experiment.

The latest MOS naturalness score approximation for this experiment will be the NISQA v1.0 model scores shown in Table 15. Here the results are not so good: instead of having problems synthesizing English using LibriTTS speakers, the problem appears in the VCTK dataset. Apparently, this model has been trained for so long in English datasets that there is not much improvement to be done in that language. Apart from that, the model seems to continue improving in the Portuguese and Basque synthesis. The scores for Spanish reported by this model are worse, though: this differs from the scores in previous VoiceMOS SSL-MOS and UTMOS-22 models done before.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*0.824±0.018*	*0.932±0.008*	*0.901±0.015*	*0.900±0.007*	*0.884±0.014*
Exp.2+SCL	0.847±0.011	0.861±0.012	**0.811±0.012**	0.877±0.029	0.871±0.025
Exp.3	0.847±0.008	0.862±0.011	0.807±0.013	0.875±0.026	0.877±0.024
Exp.3+SCL	**0.855±0.008**	**0.866±0.011**	0.807±0.015	**0.883±0.023**	**0.879±0.029**

Table 16: Speaker Encoder Cosine Similarity (SECS) after 3rd experiment.

With respect to speaker similarity, in Table 16, we can see the Speaker Encoder Cosine Similarity scores. The speaker similarity for the Portuguese language, containing only one speaker, went a little worse. It may not be much space for improvement after training for so long in this dataset containing a single speaker. In the other datasets, there has been a small improvement. The VCTK scores continue being above ground truth and improving little by little.

Concerning the problem of ”r” pronunciation at the end of sentences, in Table 17, the R-Error-Rate can be inspected. Compared with the previous experiment, the mean error rate of all the speakers in the TTS-DB dataset has been reduced, but not as much as with the previous experiment. It seems that as the training progresses, it is more challenging to reduce the error. On the other hand, something changed in this experiment: this time, the non-SCL part of the experiment had a lower error rate, and the SCL training made the score worse. In addition, there is more variability between scores in the speakers. This may be because now there are much more speakers in the whole dataset. Consequently,

Experiment	CER	SpkM1	SpkF1	SpkM2	SpkF2	SpkM3	SpkF3	Mean
Ground truth	19.9%							19.1%
Exp.2+SCL	19.6%	59.9%	64.5%	**76.8%**	95.1%	50.9%	76.9%	55.9%
Exp.3	19.1%	48.3%	**45.2%**	90.7%	93.0%	**43.0%**	**61.2%**	**44.1%**
Exp.3+SCL	19.9%	**43.9%**	62.1%	86.4%	**92.7%**	58.9%	64.9%	50.3%

Table 17: "R" at the end problem evaluation on TTS-DB dataset with the Basque language after 3rd experiment.

the model has more speakers to learn, so it cannot focus on improving specific speakers so much: TTS-DB does not have many speakers (9 speakers) compared to the newly added datasets (83 speakers). So those speakers from TTS-DB may have lost importance for the model. The CER of the synthesized sentences still remains around the ground truth CER, getting the best score we got so far of 19.1 in Experiment 3 without SCL. On the whole, the scores in this experiment have improved, but they are yet far from the ground truth scores.

4.4 Experiment 4: Continue Training

This last experiment will be a little different from the others. Here we are not going to add any new language. The purpose of this experiment is to continue training for the same languages, using the same datasets for longer, to see if the scores improve. Specifically, we are interested in seeing if the R-Error-Rate can be decreased. At the same time, we are going to see how the rest of the scores progress, thus checking if this model can be improved in general just by training for longer.

This time, we are going to compare this experiment's results with all the experiments done before and get an overview of all the models developed.

4.4.1 Experimental Setup

The experiment setup is exactly the same as in Experiment 3. The same datasets will be used, and the same hyperparameters. This time, the length of the training will be increased for the non-SCL part. The first piece of training will be trained for around 250k steps. In that training, a total of 289,066 recorded instances will be used, having a total of 42 epochs for the entire training. The second part will for the same number of steps as before: 12 epochs.

Before using the dataset available, the TTS-DB and OpenSLR-69 (Kjartansson et al., 2020) have been reviewed. Specifically, all the examples ending in an "r" character have been checked by a Speech-to-Text model, and the ones not correctly recognized have been discarded. As in the R-Error-Rate calculation, the language model of the STT model has been disabled to avoid statistical corrections unrelated to the audio. This process has been done for both the Basque and the Spanish datasets. The amount of sentences removed corresponds to a 19.19% in the TTS-DB$_{EU}$ training split, matching with the ground truth

Error Rate provided above, and none from the eval split. From the OpenSLR-76, 16.67% and 66.67% sentences have been removed from the train and eval splits. As for the Spanish datasets, from the TTS-DB$_{ES}$ training split, 5.17% of the sentences have been removed, and 5.13% from the eval split; from the ELRA-TC 49.32% and 49.27% of the sentences have been removed from the training and eval splits respectively. Listening to some of them, a few of them lack a pronunciation of the consonant, but most of them have an appreciable consonant with low energy.

4.4.2 Evaluation and Result

The first piece of training took around 7 days to complete, and the second part took around 2 days, taking a total of 9 days to complete the full training.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.08±0.09*	*4.42±0.04*	*2.46±0.10*	*3.19±0.11*	*3.16±0.16*
YourTTS	4.00±0.11	4.08±0.10	2.45±0.09		
Exp.1	4.23±0.07	4.18±0.10	**3.42±0.14**	3.04±0.09	
Exp.1+SCL	4.20±0.09	4.16±0.11	2.69±0.09	3.10±0.10	
Exp.2	4.18±0.10	4.11±0.13	2.73±0.08	3.14±0.11	2.99±0.17
Exp.2+SCL	4.21±0.08	4.12±0.09	2.77±0.10	3.09±0.13	3.06±0.14
Exp.3	4.07±0.08	4.04±0.10	2.74±0.09	3.03±0.12	3.02±0.12
Exp.3+SCL	4.19±0.09	4.06±0.10	2.78±0.09	3.14±0.12	3.16±0.16
Exp.4	**4.25±0.07**	4.16±0.11	2.88±0.10	3.20±0.12	3.12±0.14
Exp.4+SCL	4.24±0.07	**4.19±0.09**	2.91±0.09	**3.23±0.12**	**3.19±0.14**

Table 18: Mean Opinion Score (MOS) approximation by VoiceMOS 2022 baseline SSL-MOS model after 4th experiment.

In Table 18, we can see the MOS scores approximated by the VoiceMOS 2022 baseline SSL-MOS model on all the experiments so far. The ground truth refers to the MOS scores on audios from the actual dataset, and YourTTS refers to the model provided by Coqui-TTS trained in English, Portuguese, and French used for transfer learning in the first experiment. With the exception of the Portuguese language (MLS-PT), the best scores are in the model in Experiment 4. Still, the differences are small and within the confidence interval, but comparing the first with the latest experiment results, there are some improvements. In out-of-domain languages (Portuguese, Basque, and Spanish), the scores seem to remain below 4 all the time, even for the ground truth.

In Table 19, there are MOS scores provided by the model of one of the best participants of the VoiceMOS 2022, the UTMOS-22. In this model, even the in-domain (English) scores are below 4, but not in the ground truth. This may suggest that there is still room for improvement. Similarly, the latest experiment got the best results, except for the Portuguese language.

In Table 20, we can see another automatic naturalness MOS estimation, this time by NISQA v1.0. In this case, Experiment 4 got the best scores in all the datasets, including

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.03±0.05*	*4.31±0.03*	*2.79±0.08*	*3.59±0.10*	*3.62±0.15*
YourTTS	3.63±0.08	3.70±0.09	2.62±0.10		
Exp.1	3.80±0.07	3.79±0.07	**3.35±0.10**	3.13±0.08	
Exp.1+SCL	3.79±0.08	3.78±0.09	2.78±0.10	3.18±0.09	
Exp.2	3.76±0.08	3.79±0.10	2.91±0.09	3.26±0.10	3.13±0.18
Exp.2+SCL	3.80±0.09	3.74±0.07	2.94±0.09	3.20±0.11	3.18±0.15
Exp.3	3.70±0.06	3.71±0.09	3.02±0.09	3.18±0.10	3.13±.013
Exp.3+SCL	3.82±0.08	3.72±0.08	2.98±0.10	3.29±0.10	3.29±0.15
Exp.4	3.85±0.08	**3.84±0.09**	3.10±0.10	**3.36±0.10**	3.26±0.12
Exp.4+SCL	**3.87±0.07**	3.82±0.07	3.12±0.10	**3.36±0.10**	**3.31±0.14**

Table 19: Mean Opinion Score (MOS) approximation by UTMOS-22 model after 4th experiment.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*3.75±0.16*	*3.98±0.14*	*3.31±0.18*	*4.11±0.12*	*3.62±0.28*
YourTTS	3.37±0.10	3.53±0.12	3.19±0.16		
Exp.1	3.43±0.11	3.47±0.13	3.13±0.13	3.66±0.15	
Exp.1+SCL	3.42±0.10	3.43±0.13	3.12±0.13	3.65±0.15	
Exp.2	3.50±0.11	3.54±0.13	3.31±0.14	3.67±0.15	3.40±0.24
Exp.2+SCL	3.60±0.11	3.58±0.13	3.23±0.14	3.72±0.14	3.45±0.21
Exp.3	3.40±0.12	3.40±0.13	3.27±0.17	3.49±0.15	3.26±0.20
Exp.3+SCL	3.50±0.12	3.63±0.13	3.39±0.15	3.74±0.16	3.44±0.23
Exp.4	3.61±0.15	3.66±0.14	3.42±0.15	3.65±0.15	**3.56±0.25**
Exp.4+SCL	**3.74±0.11**	**3.73±0.14**	**3.46±0.13**	**3.83±0.15**	3.55±0.30

Table 20: Mean Opinion Score (MOS) naturalness approximation by NISQA v1.0 after 4th experiment.

MLS-PT. Only the MLS-PT got scores better than the ground truth, which may have some sense being a dataset trained by a non-professional with a cheap microphone, noise, and a single speaker. The other datasets have better quality in general, and it should be more difficult to reach the ground truth quality. We can compare the initial experiments with the last one to see the progress more clearly.

Experiment	**VCTK**	**LibriTTS**	**MLS-PT**	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*0.824±0.018*	*0.932±0.008*	*0.901±0.015*	*0.900±0.007*	*0.884±0.014*
YourTTS	0.845±0.009	0.858±0.011	0.803±0.010		
Exp.1	0.849±0.009	0.862±0.014	0.813±0.012	0.873±0.023	
Exp.1+SCL	0.845±0.010	**0.868±0.013**	**0.817±0.012**	0.879±0.027	
Exp.2	0.843±0.010	0.860±0.012	0.813±0.013	0.882±0.021	0.871±0.028
Exp.2+SCL	0.847±0.011	0.861±0.012	0.811±0.012	0.877±0.029	0.871±0.025
Exp.3	0.847±0.008	0.862±0.011	0.807±0.013	0.875±0.026	0.877±0.024
Exp.3+SCL	0.855±0.008	0.866±0.011	0.807±0.015	**0.883±0.023**	0.879±0.029
Exp.4	0.853±0.007	0.860±0.011	0.814±0.013	0.880±0.022	0.886±0.018
Exp.4+SCL	**0.855±0.007**	0.860±0.012	0.811±0.013	0.874±0.027	**0.890±0.021**

Table 21: Speaker Encoder Cosine Similarity (SECS) after 4th experiment.

Regarding the speaker similarity, in Table 21 we can see a comparison of the Speaker Encoder Cosine Similarity scores. In this case, there has not been an improvement. In the VCTK English dataset, there has been some improvement, but in the others, the improvement is not clear. This is reasonable, taking into account that we have trained for longer, just the language part of the training. During that training, the model has been focused on improving the audio quality and may have lost a little ability to use specific speakers. Then, the last SCL training part was completed but not extended as the first part. Maybe extending that second part of the training may alleviate the speaker-forgetting problem. Still, the scores have little difference and are within the confidence interval. In addition, for the Spanish language, the model has continued to improve a little.

To Finish, the R-Error-Rate of all the models can be checked in Table 22. This has been performed by generating 1000 audios for each speaker with sentences ending in ”r” and using a Speech-to-Text without a language model to transcribe them. The is no doubt that the ”r”-at-the-end problem has been reduced considerably just by cleaning the dataset a little and training for longer. Reaching a mean of 34.6% error rate for ”r” characters at the end is more acceptable than 79%. Probably training it for longer, the problem may be reduced even more. Nevertheless, the error rate for some speakers is still high, above 80% for speakers M2 and F2. Another aspect to notice is that the CER of the latest experiments has the best CER scores, with both parts of the experiment improving the ground truth CER.

Something additional to keep in mind is that this training length has been almost double length, and the scores have not improved much. This means that even though the model can continue improving just by training for longer, the model has more problems improving and progresses much slower. Therefore it may not be worth training longer to

Experiment	**CER**	**SpkM1**	**SpkF1**	**SpkM2**	**SpkF2**	**SpkM3**	**SpkF3**	**Mean**
Ground truth	19.9%							19.1%
Exp.1	20.8%	80.9%	94.5%	93.9%	99.4%	84.7%	95.8%	79.1%
Exp.1+SCL	19.6%	80.0%	92.2%	90.5%	96.5%	91.1%	94.0%	76.0%
Exp.2	20.0%	63.5%	78.2%	80.8%	99.4%	41.8%	88.3%	61.1%
Exp.2+SCL	19.6%	59.9%	64.5%	**76.8%**	95.1%	50.9%	76.9%	55.9%
Exp.3	19.1%	48.3%	45.2%	90.7%	93.0%	43.0%	61.2%	44.1%
Exp.3+SCL	19.9%	43.9%	62.1%	86.4%	92.7%	58.9%	64.9%	50.3%
Exp.4	18.9%	**30.7%**	48.6%	87.2%	93.4%	**31.1%**	50.6%	41.0%
Exp.4+SCL	19.6%	35.7%	**36.9%**	84.5%	**86.6%**	31.7%	**47.3%**	**34.6%**

Table 22: ”R” at the end problem evaluation on TTS-DB dataset with the Basque language. The ground truth row represents the Speech-to-Text overall performance in the dataset. The Mean represents the mean between all the speakers, not just the 6 included here.

get a small improvement, depending on the case.

Summarizing in general, adding languages does not seem to make the model worse in the previous languages. On the contrary, adding languages seems to improve the speech quality of the model, even though it is just a little and not significant.

4.5 Human Evaluation

The subjective human evaluation was performed by 16 Basque and 24 Spanish native speakers. For all the tests, both the speech naturalness and the Sim-MOS for speaker similarity have been calculated. Additionally, each of the experiments has been performed in two modalities: dataset speakers and zero-shot. First, the dataset speaker results will be shown, and these results will be comparable with the results shown above for each experiment. The zero-shot speakers have been performed using external speakers from the ZureTTS project (Erro et al., 2014, 2015).

Experiment	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.779±0.082*	*4.790±0.068*
Exp.1+SCL	3.699±0.181	
Exp.2+SCL	3.750±0.150	3.770±0.148
Exp.3+SCL	3.838±0.167	3.830±0.133
Exp.4+SCL	**3.926±0.157**	**3.990±0.136**

Table 23: Mean Opinion Score (MOS) on naturalness by Human Evaluators.

In Table 23, we can see the subjective naturalness of generated speech in Basque and Spanish. In both cases, the scores improved as we completed each experiment, ending up with scores of almost 4 in naturalness, but still far from the ground truth scores.

Similarly, in Table 24, the speaker similarity scores are shown. In this case, we have a couple of scores above 4, so we can conclude that the speaker imitation is quite good. It is

Experiment	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.338±0.205*	*4.475±0.121*
Exp.1+SCL	3.993±0.188	
Exp.2+SCL	3.993±0.180	4.020±0.145
Exp.3+SCL	3.956±0.200	3.995±0.149
Exp.4+SCL	**4.015±0.190**	**4.170±0.150**

Table 24: Mean Opinion Score (MOS) on speaker similarity by Human Evaluators.

still not as good as the ground truth, but the difference is not so big as with naturalness. Comparing each experiment, the model scores also improve as we do more training.

As for zero-shot speakers, in Table 25, the naturalness scores with out-of-domain speakers are shown. Surprisingly, the difference between ground truth and the experiment scores is similar. Therefore, using unknown speakers does not affect the quality of the speech produced.

Experiment	**TTS-DB$_{EU}$**	**TTS-DB$_{ES}$**
Ground truth	*4.740±0.122*	*4.795±0.091*
Exp.1+SCL	3.604±0.213	
Exp.2+SCL	3.500±0.208	3.330±0.189
Exp.3+SCL	3.750±0.218	3.473±0.188
Exp.4+SCL	**3.850±0.223**	**3.964±0.167**

Table 25: Mean Opinion Score (MOS) on naturalness by Human Evaluators with zero-shot speakers.

As for speaker similarity with zero-shot, the scores are similar, and later experiments seem to be better, as it is shown in Table 26. But in this case, for the Basque language, Experiment 4 performed worse than Experiment 3. In the Spanish language, there is much more progress between experiments, and the latest experiment score is almost the same as in-domain scores. Overall, there is a positive trend as the training length is extended. Otherwise, the scores here are worse than the ones with dataset speakers in Table 24, meaning that it is more challenging for the model to learn to generate speech with unknown speakers.

Altogether, with the human evaluation results, we can reach the same conclusion as with automatic evaluation: the results improve as we keep adding datasets and training for longer. This may imply that these models are undertrained and better scores can be reached just by adding more training time, as we have seen with Experiment 4. Notwithstanding, these evaluation has been completed with a limited number of evaluators. This can be appreciated in the big confidence interval of the results, meaning that there is high variability in the scores.

Experiment	**TTS-DB**$_{EU}$	**TTS-DB**$_{ES}$
Ground truth	*4.708±0.128*	*4.562±0.177*
Exp.1+SCL	3.354±0.228	
Exp.2+SCL	3.375±0.200	3.455±0.210
Exp.3+SCL	**3.396±0.211**	3.750±0.196
Exp.4+SCL	3.302±0.220	**4.009±0.173**

Table 26: Mean Opinion Score (MOS) on speaker similarity by Human Evaluators with zero-shot speakers.

4.6 Discarded Experiments

Before deciding to do Experiment 4, we did some tests with other experiments. We hypothesized that some languages, like French, might influence the ”r” character pronunciation in Spanish. We also found some recording examples where the pronunciation of the ”r” at the end was not very clear. So we designed a set of three experiments, done in the following order:

1. *Experiment 1.1 removing French*: This consisted in training the model on Experiment 1, using the final model provided by Coqui-TTS for transfer learning, which had already been trained on a French dataset. However, during the training, we intentionally removed the French dataset in order for the model to forget that language. Then, as in Experiment 1, we trained the model on the Basque dataset for it to learn the language. The setup for the training is exactly the same as for the original Experiment 1.

2. *Experiment 1.2 without French*: Similar to the previous approach, but doing transfer learning from a model that has not been trained on a French dataset before. For that, we did transfer learning from YourTTS model Experiment 2 (Casanova et al., 2022b). The setup for the training is again the same as for the original Experiment 1.

3. *Experiment 1.3 cleaning up the dataset*: This reproduces Experiment 1, but cleaning up the Basque dataset from recordings with a not-so-clean ”r” pronunciation at the end. This is the cleaned dataset used for Experiment 4: check Section 4.4 for the details of how the filtering has been performed. Again the training was performed maintaining the same setup as for Experiment 1.

In table 27, we can see the R-Error-Rate of the discarded models compared with the final Experiment 1 and our latest experiment, Experiment 4. We did this experiment to decide how to solve the ”R”-at-the-end problem. There is no doubt that removing the French dataset helped the most (Exp.1.1), even more than training on a model that has not been trained before with the French language (Exp.1.2). Training on a cleaned-up dataset, removing sentences with unclear pronunciations of the consonant also helped.

Experiment	**SpkM1**	**SpkF1**	**SpkM2**	**SpkF2**	**SpkM3**	**SpkF3**	**Mean**
Ground truth							19.1%
Exp.1	80.9%	94.5%	93.9%	99.4%	84.7%	95.8%	79.1%
Exp.1.1	**61.8%**	**71.9%**	92.6%	**95.4%**	**40.3%**	**83.0%**	**66.0%**
Exp.1.2	65.3%	87.5%	**83.5%**	96.4%	71.0%	86.0%	75.9%
Exp.1.3	73.0%	89.7%	88.6%	98.7%	71.7%	95.3%	72.6%
Exp.4+SCL	**35.7%**	**36.9%**	84.5%	**86.6%**	**31.7%**	**47.3%**	**34.6%**

Table 27: R-Error-Rate scores on TTS-DB dataset with the Basque language on the discarded experiments, comparing them with the first and latest experiments of our final models.

Unfortunately, the problem did not disappear completely, not even in Experiment 1.2, where the model had not seen the French language before. Additionally, all the scores are still too high: 66% is the best-reached score, being the problem far from disappearing. So we discarded the idea of French being the primary source of the problem, considering the possibility of being a limitation of the model. The outcome obtained here led us to decide to extend the training, as explained in Experiment 4, with the cleaned version of the dataset and see if that forced the model to continue learning. The idea of extending the last experiment was for the dataset to include examples where a Basque-style pronunciation of ”r” was correct and emphasize learning the new languages added here, as Basque, Spanish, Galician, and Catalan all of them use those phones frequently. This has the additional benefit of not losing the ability to synthesize speech with the French language.

4.7 Chapter Summary

In this chapter, we have described all the experiments performed to train the Text-to-Speech model in detail. We started doing transfer learning from a multi-language model pre-trained in English, Portuguese, and French. Our first experiment added the Basque language, the second experiment the Spanish language, the third experiment two languages, Catalan and Galician, and the latest experiment extended the training for the model to continue improving. We also evaluated the model in each of the steps along the way. We showed that even though we continue adding languages, there is no proof of the model finding difficulties learning or decreasing the performance on the previous languages. Indeed, there are signs that indicate that, as we continue adding languages and extending the training time, the model continues improving slightly.

5 Application

In this chapter, the creation of a Speech-to-Speech model for machine translation with voice conversion will be described. In other words, this describes a deliverable application that can translate your voice to another language. The main focus of the research has been to develop a good multilingual Text-to-Speech model. Along the way, the need to have good Speech-to-Text transcription models has arisen in order to be able to evaluate and debug some problems of the created models. To take advantage of the created models, a small prototype has been created that brings together all the models built here plus others obtained from the community.

The prototype first uses Speech-to-Text models to transcribe audio taken as input. Some of these models have been obtained from the community, and two of them have been created in-house. Subsequently, the model uses Machine Translation to convert the transcribed text to the target language. Machine Translation models are out of the scope of this research, so they have been used in inference without training. Finally, the Text-to-Speech model is used to synthesize an output waveform with the new language. All the models involved here support the same languages: English, Basque, Spanish, Catalan, Galician, Portuguese, and French. The inner details of each of the models are described below.

5.1 Text-to-Speech Module

The Text-to-Speech module includes all the models developed in the different experiments performed during this research, which are the following:

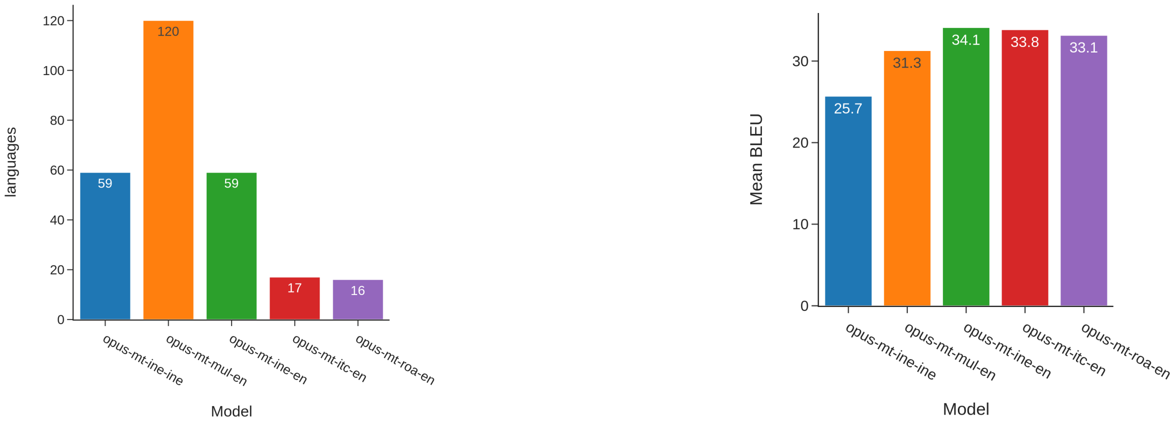
- `Experiment 1 EN-PT-FR-EU`: English, Portuguese, French, and Basque support.
- `Experiment 1 EN-PT-FR-EU SCL`: English, Portuguese, French, and Basque support focused on speaker similarity.
- `Experiment 2 EN-PT-FR-EU-ES`: All of the languages above, plus Spanish.
- `Experiment 2 EN-PT-FR-EU-ES SCL`: All of the languages above, plus Spanish, focused on speaker similarity.
- `Experiment 3 EN-PT-FR-EU-ES-CA-GL`: All of the languages above, plus Catalan and Galician.
- `Experiment 3 EN-PT-FR-EU-ES-CA-GL SCL`: All of the languages above, plus Catalan and Galician, focused on speaker similarity.
- `Experiment 4 EN-PT-FR-EU-ES-CA-GL`: All of the languages above trained for longer.
- `Experiment 4 EN-PT-FR-EU-ES-CA-GL SCL`: All of the languages above, training for longer, including speaker similarity loss.

This module supports voice conversion from an input audio file for zero-shot or using any speaker from the dataset. This audio input file, when provided, is also normalized following the techniques explained in Section 3.2.3. Furthermore, the text passed to the TTS modules is pre-processed using the *text-normalizer* package described in section 3.2.2.

5.2 Machine Translation Module

The Machine Translation inference module, by default, uses multiple Opus-MT models from the University of Helsinki and CSC (Tiedemann et al., 2022), with the alternative of using Google Translate API mainly for testing.

The Opus-MT inference technique uses one or multiple Opus-MT models to translate between the source language to the target language: if it exists a model for direct translation, for example, from Portuguese to Spanish, it uses it. If such a model does not exist, it translates first to English and then from English to the target language. Moreover, if there is no direct lang-to-English or English-to-lang translation model and it needs to be used, it uses the multilingual `opus-mt-mul-en` or `opus-mt-en-mul` models, respectively.

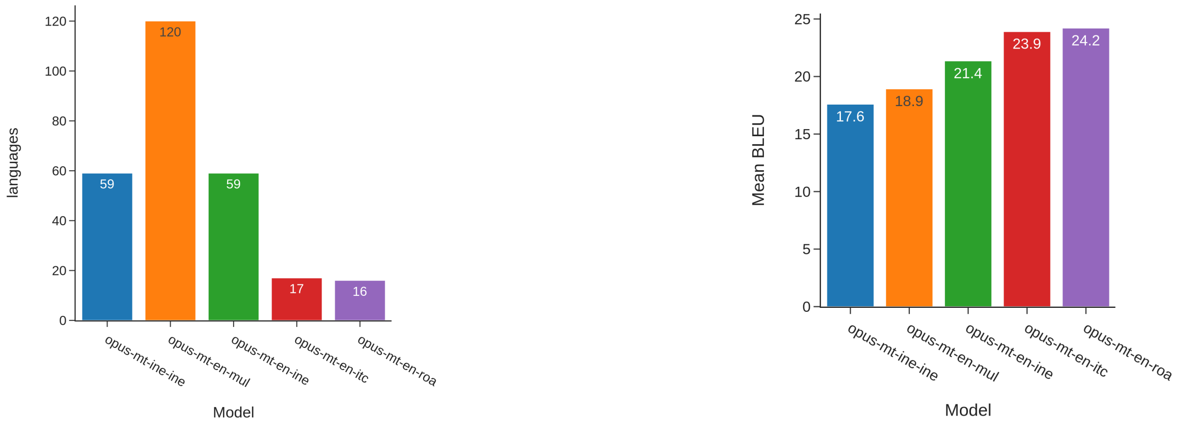


(a) Lang-to-English models languages.

(b) Lang-to-English models BLEU scores.

Figure 23: Machine Translation models to translate from multiple languages to English.

The opus-MT project (Tiedemann et al., 2022) trained various multilingual machine translation models with different language support. In Figure 23, we can see the differences between multiple languages to English multilingual models: the number of languages supported and their BLEU score. The BLEU score is a metric to evaluate machine-translated text: a value of 100 is the best score, and 0 is the worst. Similarly, in Figure 24, the differences between multilingual models to translate from English to multiple languages can be seen. In both cases, the `opus-mt-mul-en` and `opus-mt-en-mul` are the ones to support more languages. At the same time, their score is not the best, but there is not so much difference from the top scoring models.



(a) English-to-Lang models languages.

(b) English-to-Lang models BLEU scores.

Figure 24: Machine Translation models to translate from English to multiple languages.

For our application, we decided to use the `mul` models for simplicity. In the future, a better algorithm can be designed that chooses the model supporting the language with the best score. Another possibility is to fine-tune the models provided with more and better quality text on the languages required. All this has been left for a future project.

5.3 Speech-to-Text Module

For the Speech-to-Text, we decided to use multiple Deep Speech models (Hannun et al., 2014a) because it is a lightweight and well-known architecture, with multiple implementations and models shared by other researchers in different languages.

In Figure 25, we can see a simplified diagram of the Deep Speech model created by the Baidu research team. The system is formed by six layers, having Mel-frequency cepstral coefficients (MFCC) as input and text as output. The first three layers are formed by Equation 15, with simple rectified-linear activation functions (ReLU) and its own weight matrix (W^l) and bias (b^l). The fourth layer is a bidirectional recurrent neural network: specifically, an LSTM in Mozilla's implementation that is used here. The fifth layer is another ReLU layer but takes both the forward and backward units of the LSTM as input. The last layer is a softmax function that gives the predicted character probabilities for each time step.

$$h_t^l = \text{ReLU}(W^l h_t^{l-1} + b^l) \tag{15}$$

Besides, often an n-gram Language Model (LM) is attached to the output to add word knowledge and improve the neural model character output. This language model is usually trained on a huge text corpus beforehand. Equation 16 is used to find a balance between the recurrent neural network probabilities output, the language model statistics, and the

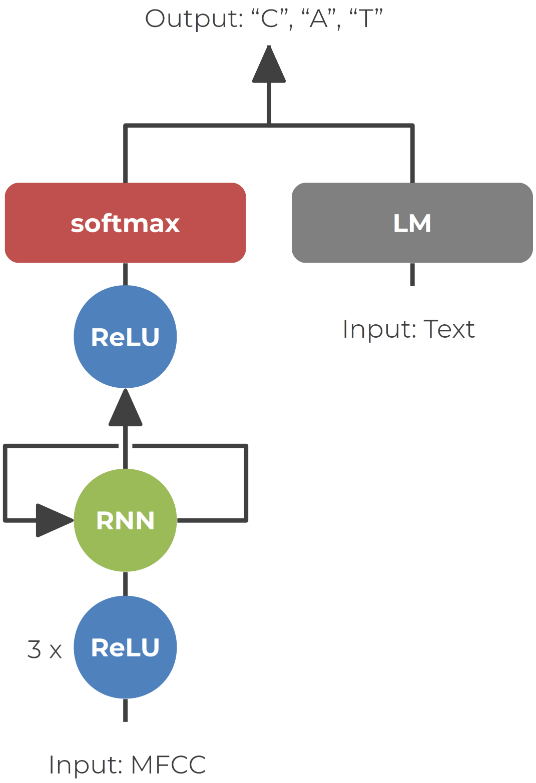


Figure 25: Deep Speech Speech-to-Text model architecture.

length of the sentence. $P_{NN}\cdot)$ referes to the neural network probability and $P_{LM}(\cdot)$ to the language model probability; x is the audio features input and o the text candidate output. The α and β hyperparameters must be adjusted for each language and trained model. This equation is maximized using the beam search heuristic algorithm (Hannun et al., 2014b) that searches for a good-enough solution by expanding the most promising candidates.

$$Q(\mathbf{o}) = \log(P_{NN}(\mathbf{o}|\mathbf{x}) + \alpha\, P_{LM}(\mathbf{o})) + \beta\, \text{word\_count}(\mathbf{o}) \tag{16}$$

Table 28 lists all the Deep Speech models used by the application developed here. The Basque and Galician STT models have been trained for this project, and the detailed process is described below. The Basque Deep Speech model was trained to improve the currently available model: we needed a more accurate model to evaluate our Text-to-Speech models. The Galician model was not available before, so it was trained from scratch. The English model was trained by the Coqui Team [10], the Portuguese model by Francis Tyers and the Inclusive Technology for Marginalised Languages (ITML) [11], the French model by the Common Voice FR project and revived by Waser Technologie[12], the Spanish model by the by Danber and released under the Jaco-Assistant project[13], and the Catalan model by Ciaran O'Reilly [14]. During inference, when using theses models in the application, the speech signal is normalized as explained in Section 3.2.3 before passing it to the specific STT model.

Language	Version	Creator	CER	WER
English	v1.0.0-huge-vocab	Coqui	4.50% / 13.60%	1.60% / 6.40%
Portuguese	v0.1.1	ITML	73.20%	26.70%
French	v0.9	commonvoice-fr	31.50%	15.20%
Basque	v0.1.8	Xabier	10.65%	4.21%
Spanish	v0.0.1	Jaco-Assistant	16.50%	7.60%
Galician	v0.1.3	Xabier	16.38%	6.83%
Catalan	v0.14.0	Ciaran O'Reilly	13.29%	-

Table 28: Deep Speech models used in this project for user speech transcription and their scores in the Common Voice dataset test split. The Basque and Galician models have been trained in this project. The English model has been tested in Librispeech clean and other splits instead of Common Voice.

The Basque and Galician models have been trained following Francis M. Tyers and Josh Meyer's approach to training Speech-to-Text models for minority languages with scarce resources (Tyers and Meyer, 2021). As audio and transcriptions training data, different versions of the Common Voice dataset (Ardila et al., 2019) have been used. In table 29,

[10] https://coqui.ai/english/coqui/v1.0.0-huge-vocab

[11] https://coqui.ai/portuguese/itml/v0.1.1

[12] https://coqui.ai/french/commonvoice-fr/v0.9

[13] https://coqui.ai/spanish/jaco-assistant/v0.0.1

[14] https://coqui.ai/catalan/ccoreilly/v0.14.0

we can see the different datasets used here for training. Something to take into account is that the Galician dataset does not have many speakers, so it has been more challenging to train.

Language	**Autonym**	**Locale**	**CV**	**Training**	**Audio**	**Clips**	**Speakers**	**|V|**
Basque	Euskara	`eu`	6.1	9:53:04	10:51:34	7,505	53	28
Basque	Euskara	`eu`	12.0	-	15:50:01	10,905	64	28
Galician	Galego	`gl`	10.0	2:25:07	4:51:08	3,403	6	30
Galician	Galego	`gl`	12.0	3:17:12	6:17:56	5,008	8	30

Table 29: Datasets used to train the Speech-to-Text models. The Training refers to the time required to train the initial models without counting LM tweaks, and |V| refers to the number of symbols in the alphabet, defining the size of the softmax layer.

All the different versions of the Speech-to-Text models trained here can be downloaded from the Aholab team web server [15].

5.3.1 Basque STT Model

The Basque STT model has been trained, continuing Francis M. Tyer's previous work (Tyers and Meyer, 2021). For the hyperparameters, we started from their best results: learning rate of 0.001, dropout of 0.2, SpecAugment enabled (Park et al., 2019), and for 100 epochs. SpecAugment is a simple data augmentation method frequently used in speech recognition tasks where the blocks of frequency channels and time steps are masked in the input features. The starting LM alpha and beta hyperparameters were 1.3388886030877378 and 4.8289231615211 in the beginning. This model was used not only for the prototype but also to calculate the R-Error-Rate in Section 3.4.

In Table 30, we can see the different versions trained and their scores. For the neuronal part of the Deep Speech model, the Common Voice versions 6.1 and 12.0 has been used. In versions `v0.1.1` and `v0.1.2`, we just tried to reproduce previous results (Tyers and Meyer, 2021) but updating the underlying CUDA toolkit from version 11 to version 12: this improved the model slightly. The Language Model was trained using Common Voice transcriptions (Ardila et al., 2019) and the Opus (Tiedemann, 2012) corpora with modified Kneser-Ney smoothing (Heafield, 2011; Heafield et al., 2013). The version `v0.1.3` was an improvement of the Language model by adding a recent Wikipedia dump: this reduced the model errors considerably. In version `v0.1.4`, we optimized the language model by searching for better alpha and beta hyperparameters during 24 hours using Optuna hyperparameter optimization framework (Akiba et al., 2019), producing a small improvement in the final model. For version `v0.1.5`, we added the EusCrawl corpus (Artetxe et al., 2022) to the language model, improving the model a little more but removing the Wikipedia corpus. This is because the EusCrawl corpus already includes the Wikipedia corpus, avoiding unnecessary data duplicity, which can extend the training without much improvement.

[15] https://aholab.ehu.eus/∼xzuazo/models/

Version	CV	LM Corpora	LM Size	WER	CER
v0.1.1	6.1	-	-	68.55%	15.60%
v0.1.2	6.1	CV, Opus	234M	17.09%	5.99%
v0.1.3	6.1	CV, Opus, Wikipedia	309M	15.87%	5.63%
v0.1.4	6.1	CV, Opus, Wikipedia	309M	14.53%	5.28%
v0.1.5	6.1	CV, Opus, EusCrawl	590M	14.31%	5.23%
v0.1.6	6.1	CV, Opus, Wikipedia, EusCrawl	638M	14.12%	5.18%
v0.1.7	12.0	CV, Opus, Wikipedia, EusCrawl	638M	12.00%	4.48%
v0.1.8	12.0	CV, Opus, Wikipedia, EusCrawl	638M	**10.65%**	**4.21%**

Table 30: Basque Deep Speech model versions and their Character Error Rate (CER) and Word Error Rate (WER) reported on the test set. Model versions v0.1.1 and v0.1.2 were previously trained by Francis M. Tyers (Tyers and Meyer, 2021) in CUDA 11, here they were retrained on CUDA 12: this made the scores improve slightly. For model version v0.1.4, LM optimization was done for 24 hours. For model version v0.1.8, full LM optimization was performed for 2400 trials.

On version `v0.1.6`, we used both EusCrawl and the latest Wikipedia dump to check the differences. The model improved slightly, but not much. For version `v0.1.7`, we update the Common Voice version from 6.1 to 12.0. As we can see, updating the Common Voice dataset version leads to a great score. However, between Common Voice versions, there is no guarantee to maintain the same splits, and they are not guaranteed to be comparable. In version `v0.1.8`, we just optimized the language model, but this time for 2400 trials, which took around 24 days to complete. The best alpha and beta values found in the latest version of the model were 1.4428895547940739 and 4.999123396032508, respectively, reducing the final WER value by 1.35%. Optimizing the language model takes time but can reduce the final scores even more without the need to have a bigger dataset or create a more convoluted model.

Anyway, in these final transcription models, the results are good enough to get an idea of the approximate error rates of each of the Text-to-Speech models generated previously. As we can see, updating the Common Voice dataset versions seems to reduce the errors significantly, making clear the need to increase the audio dataset of languages with few resources. However, between Common Voice versions, there is no guarantee to maintain the same splits, and they are not guaranteed to be comparable. But the results shown here are good enough to get an idea of the approximate error rates of each model individually.

Version	CV	LM Corpora	LM Size	WER	CER
v0.1.7-OnlyWiki	12.0	CV, Wikipedia	70M	18.66%	6.26%
v0.1.7-OnlyEusCrawl	12.0	CV, EusCrawl	344M	**15.76%**	**5.54%**

Table 31: Basque Deep Speech model versions using only the Wikipedia or the EusCrawl corpus for the language model.

As a side dataset ablation test, we also re-trained the latest version of the model's language model, keeping only the Common Voice transcriptions and Wikipedia for the first test and the EusCrawl for the second test. The results can be seen in Table 31. This experiment was performed to make a fair comparison between Wikipedia dump and EusCrawl as a corpus to train the language model. Considering that these models should have almost no redundancy in the data, we can use their results to compare the quality of both datasets fairly. Starting from the same Deep Speech neural model, it is clear that the EusCrawl dataset achieved better scores in exchange for increasing the language model size. No doubt, the EusCrawl brings more linguistic richness to the model and is a good source for training language models.

5.3.2 Galician STT Model

For the Galician STT model, we did not find any previously trained Deep Speech model. To tackle this difficulty, we did a first hyperparameter sweep (Tyers and Meyer, 2021) process consisting of 18 short experiments, testing the following parameter values: learning rate in the range 0.001, 0.0001, 0.00001, values for dropout in the range 0.2, 0.4, 0.6 and whether to enable SpecAugment (Park et al., 2019). The best scoring hyperparameters were: learning rate of 0.00001, dropout of 0.2, and SpecAugment disabled.

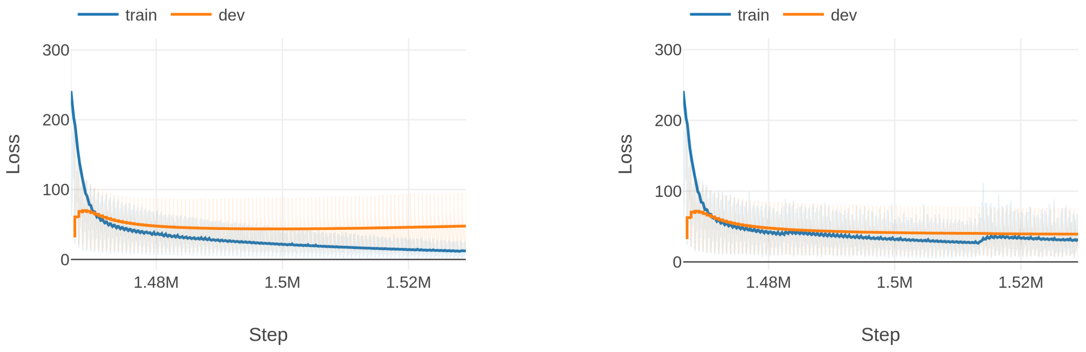
Version	CV	LM Corpora	LM Size	WER	CER
v0.1.0	10.0	CV, Opus, Wikipedia	398M	21.90%	9.10%
v0.1.1	12.0	CV, Opus, Wikipedia	473M	18.37%	7.66%
v0.1.2	12.0	CV, Opus, Wikipedia	473M	16.42%	6.85%
v0.1.3	12.0	CV, Opus, Wikipedia, SLIGalWeb	527M	**16.38%**	**6.83%**

Table 32: Galician Deep Speech model versions and their Character Error Rate (CER) and Word Error Rate (WER) reported on the test set. Model version v0.1.2 and the following were trained with SpecAugment enabled.

In table 32, we can see the different experiments performed. In the first version, `v0.1.0`, we trained it in Common Voice version 10.0 for the Deep Speech neural model. For the language model, Common Voice transcriptions, Opus, and Wikipedia were used. For version `v0.1.1`, we updated the Common Voice dataset from version 10 to version 12, which brought some improvement to the model. Then, in version `v0.1.2`, we show the model started to have some overfitting after the update, so we enabled SpecAugment again, improving the results even more. In Figure 26, both the training plots can be seen, with and without overfitting.

Last but not least, we added the SLI GalWeb corpus (Agerri et al., 2018), composed of crawled texts from various domains by the IXA pipes tools (Agerri et al., 2014). This last corpus improved the scores a little more, getting the best model of all.

As a final note, there is an evident lack of public resources for the Galician language both in audio and in text corpora, and this made the model not reach such good results. Still,



(a) SpecAugment off

(b) SpecAugment on

Figure 26: Galician model versions `v0.1.1` and `v0.1.2` trained first without and then with SpechAugment, probing that enabling it fixed the overfitting problem as the audio dataset size increased. The blue lines refer to the training loss, and the orange lines to the validation loss.

with the limited resources available, the results reached can be considered good enough for some applications, such as the prototype presented here.

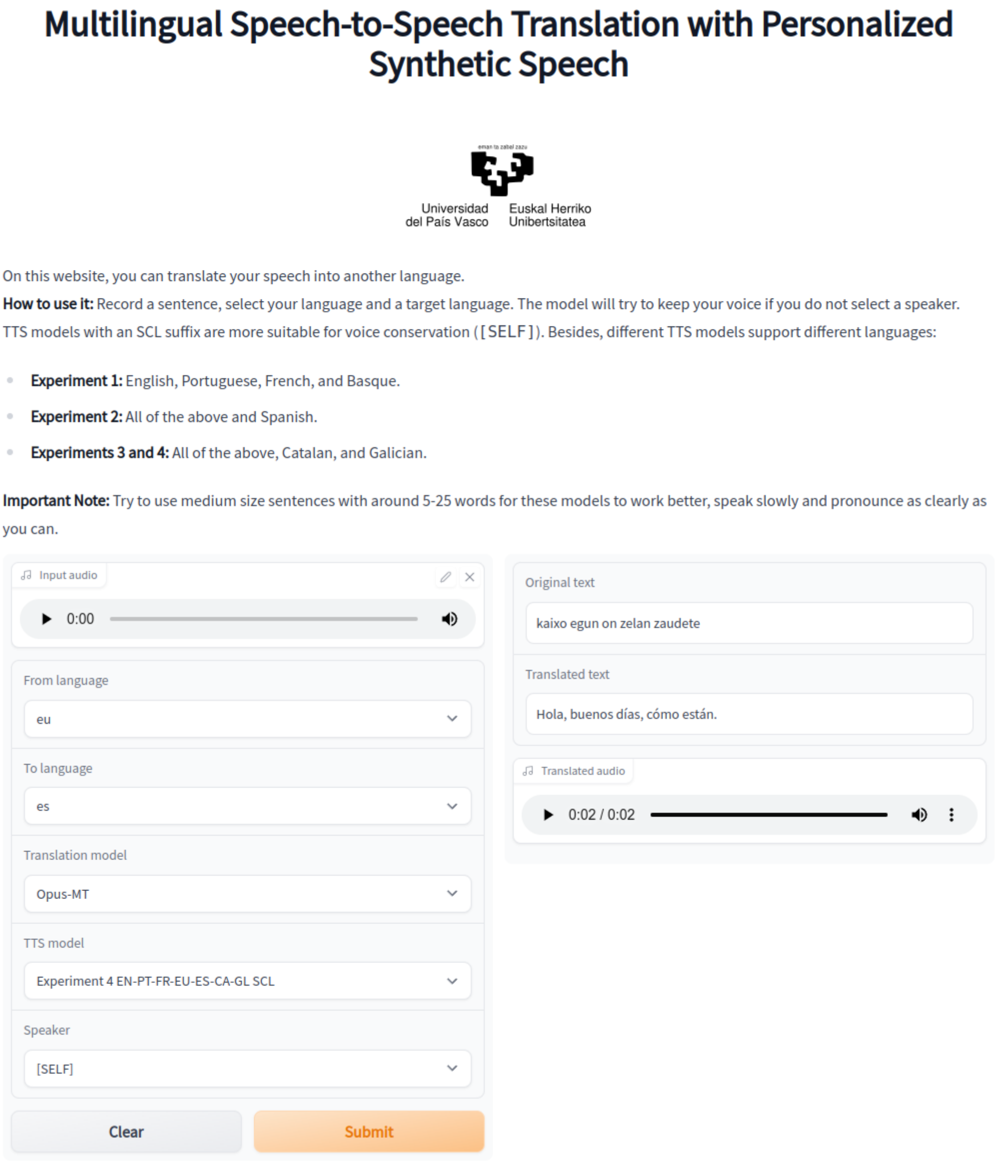
5.4 Online Web Interface

The final prototype has a modular approach, having a separate module for each task. The main idea is to be reusable for future models of Text-to-Speech, Machine Translation, Speech-to-Text, or even End2End models. The web user interface has been created using the Gradio Framework[16] following this lowly-coupled methodology and has been tested to work both on desktop and mobile devices. In addition, the different tasks supported can be independently enabled or disabled. The prototype has been deployed in Aholab research group servers with the Speech-to-Speech Translation module enabled and can be thoroughly tested online [17].

In Image 27, the S2ST web interface can be appreciated. The user records a sentence and selects the source and the target language. After pushing the submit button, the speech will be translated into another language. By default, the speaker voice will be conserved (`[SELF]`), but there is a drop-down menu with a list of speakers to choose from if preferred. Additionally, it also has options to select other synthesis models to try, which for now include the different experiments carried out here. Machine translation models include the Opus-MT models, but there is an option to use the Google Translate API too.

[16] https://gradio.app/

[17] https://aholab.ehu.eus/S2S/



Multilingual Speech-to-Speech Translation with Personalized Synthetic Speech

On this website, you can translate your speech into another language.

How to use it: Record a sentence, select your language and a target language. The model will try to keep your voice if you do not select a speaker. TTS models with an SCL suffix are more suitable for voice conservation ([SELF]). Besides, different TTS models support different languages:

- **Experiment 1:** English, Portuguese, French, and Basque.
- **Experiment 2:** All of the above and Spanish.
- **Experiments 3 and 4:** All of the above, Catalan, and Galician.

Important Note: Try to use medium size sentences with around 5-25 words for these models to work better, speak slowly and pronounce as clearly as you can.

From language: eu

To language: es

Translation model: Opus-MT

TTS model: Experiment 4 EN-PT-FR-EU-ES-CA-GL SCL

Speaker: [SELF]

Original text: kaixo egun on zelan zaudete

Translated text: Hola, buenos días, cómo están.

Figure 27: Speech-to-Speech Translation model prototype web interface.

5.5 Chapter Summary

In this chapter, we have seen an example of application development using our Text-to-Speech models. The prototype described here does Speech-to-Speech Translation with voice conversion. The model is structured by attaching three models in cascade: Speech-to-Text module, Machine Translation module, and Text-to-Speech module. For the Speech-to-Text module, we used Deep Speech models, one for each language. For that, we trained two of them in-house, the Basque and Galician models, and the training process has been detailed here. Next, Machine Translation models were used in inference together, choosing between different models to make inferences from depending on the source and target language. Finally, all the models were put together in a web user interface.

6 Discussion and Future Work

In this research project, we have worked with multiple related speech and language processing technologies, using Text-to-Speech, Speech-to-Text, and Machine Translation models, developing and evaluating some of them on the way. Here we will summarize some of the results and try to add some deeper insights into the quality of the trained models. First, we are going to review the outcome of each of the experiments. Then, we are going to mention some limitations of the work here. Finally, we will propose possible future work to alleviate the current weaknesses and continue improving the multilingual Text-to-Speech models presented.

6.1 Discussion

In this work, we started from a TTS multilingual model supporting three languages and added Spanish, Basque, Catalan, and Galician support to it. We did this step by step: adding Basque support first, then Spanish, and finally Catalan and Galician. Initially, we were interested in whether incorporating more languages into the multilingual model decreases its performance or if the model found more difficulties in learning the new languages as we progressed through the experiments. In Chapter 4, we saw the evaluation results of each of the steps in detail. We evaluated each model from the speech naturalness and speaker similarity points of view using various evaluation methods. In addition, we found that some of the models have difficulties synthesizing the ”r” character at the end of the sentences, so we also evaluated the R-Error-Rate metric.

As already explained, in the *Experiment 1* described in Section 4.1, we added Basque support to the model and compared it with the original model, only supporting English, Portuguese, and French. Checking the naturalness of the language, our model improved the scores from the original model in most of the evaluated metrics. The improvement was minute but generalized along all the tests. The speaker similarity scores got better results than the original YourTTS, even better than the ground truth for some of the datasets tested. The general result of the model was very positive.

Unfortunately, the RER score was very high: the model had problems pronouncing sentences ending in the ”r” character. We also tested the rest of the characters from the Basque alphabet using a Text-to-Text speech model, and this problem does not seem to be present or is very infrequent in sentences with other endings. Anyway, these models have trouble synthesizing the voiced alveolar tap [ɾ] and the voiced alveolar trill [r] phones from Basque. The reason for this limitation has not been found, but other phones from other languages not very present in English could also cause complications. For the problem in Basque, we created the R-Error-Rate metric to trace this problem along the rest of the trained models. For this first experiment, the RER at the end of the sentences was 76%.

In *Experiment 2*, presented in Section 4.2, we just added another language: Spanish. One of the datasets used to train this model was shared with the previous experiment. This meant that only one new speaker was added to the training dataset; the others were already learned while learning the Basque language. The naturalness scores of this model also got a modest improvement over the previous model. The improvement was not completely clear because the differences were within the statistical variance intervals. But a slight generalized improvement can certainly be appreciated across all the languages tested. Nonetheless, the improvement in speaker similarity was not clear for the English and Portuguese languages. This may be due to the new dataset adding only one new speaker, meaning that to obtain better speaker similarity results, adding more speakers is recommended. On the other hand, the R-Error-Rate scores improved a lot, suggesting that adding datasets and training for longer can reduce the problem. Still, the error rate was relatively high, around 56%. This training confirmed that adding languages is not worsening the quality of the model, nor is it finding it difficult to learn new languages.

As for *Experiment 3*, detailed in Section 4.3, we added two languages instead of one to see if the model can learn multiple languages simultaneously. The languages were Galician and Catalan, which contain common linguistic roots so as not to stress the problem too much. The datasets added for these languages were from a different source, adding around 80 new speakers to the dataset. The naturalness scores did not improve for the English dataset. For the Basque and Spanish languages, it seems to be a slight improvement, but it is not as clear as in the previous experiments. With respect to the speaker similarity, it was a little better than before. However, the progress is not huge, meaning that there is not so much room for improvement. The RER continued decreasing, reaching an error rate of around 44-50%: the reduction was smaller than before. In general, the improvement of the models as we keep adding languages seems to decrease, but there are no signals of model deterioration in any sense. In other words, the models do not seem to encounter any difficulty in learning new languages and speakers.

With regard to *Experiment 4*, described in Section 4.4, we did not add any new language this time. We just continued training in the same languages for longer. The purpose of this piece of training was to see if the scores can continue improving just by training and if the R-Error-Rate is reduced. This model was left training for a little longer than the previous experiments, about two days more. Checking the speech quality, the model got the best scores, comparing it with the rest of the experiments. The improvement between each of the experiments is not so clear, but by checking the whole progress, the models seem to improve little by little and achieve better performance over time. The speaker quality did not improve here, which may make some sense because we did not add any new speakers. Therefore, the addition of new voices is a requirement to improve the model on speaker similarity. As for the R-Error-Rate, it was reduced to 35%. Not bad, considering that we had an R-Error-Rate of 76% in the first experiment and the ground truth error rate is around 19%.

After finishing all the experiments, we also performed various human evaluation tests to judge the different experiments from a subjective point of view. All in all, the same conclusions can be drawn from the human MOS as from the automatic tests: As we continue to train the models and add languages, the models do not degrade and seem to improve. In these tests, we also performed zero-shot evaluation using speakers from an external dataset not previously seen by the model. In this zero-shot scenario, the naturalness of the speech produced is not affected, but the speaker conversion is more difficult for the model to perform, obtaining slightly worse scores. Nevertheless, the score differences fall between the confidence interval, so the difference cannot be considered significant on our tests. Hence, we can conclude that the created Text-to-Speech models are appropriate for use in real-life scenarios with any speaker's voice.

To summarize, as we keep adding new datasets with new languages and speakers, the model not only does learn those new languages and speakers but also gets higher scores and produces better results in general. We also found that as we continued training the model, the improvements were smaller and more challenging.

6.2 Limitations

The "r" at the end problem gets better as we progress in the experiments and extend the training, but it does not disappear completely. This indicates that this new model may have problems synthesizing some phones that are not present in the most common languages used to train Text-to-Speech models.

For the evaluation systems, we used different models that approximate speech naturalness mean opinion score values. This metric is usually performed by human beings and is entirely subjective. We are unsure about the trustfulness of these scores produced automatically by deep neural models. Mainly because these MOS models are frequently trained just in the majority languages like English; so using them with out-of-domain languages, their scores may not accurately reflect reality. A better approach is to use a big crowd to score the audio files, but this is expensive and requires time. In any case, unlike in other deep learning fields, in speech synthesis, there is a clear lack of objective metrics to evaluate the quality of the generated speech.

6.3 Future Work

After successfully training a multilingual Text-to-Speech model step by step that supports English, Basque, Spanish, and other languages, the next steps may be to improve the current model or add more languages support. There is no doubt that to achieve both of those things, having good resources is a must. Consequently, creating more and better datasets for the languages involved here can make the model become better, and creating resources for other languages can help in adding new language support. As it is widely known, speech resources with parallel audio and transcriptions are abundant in English but not so much in other languages. Especially for minor languages like Basque and Galician, we found a lot of problems in finding good resources and datasets. Without the resources shared by the Aholab team, this would have been much more complicated, if not impossible.

With respect to the specific experiment completed here, more model hyperparameter tuning may help the model improve even more. Also, trying different strategies could help, like adding more languages at the same time, training for longer, or increasing the batch size, which was reduced here due to hardware constraints. The latter is important for the model attention modules to train correctly and achieve better results.

Another possible area of research is the creation of better evaluation techniques. As dictated above, the current automatic naturalness and speaker similarity metrics may not reflect speech real quality. Another common approach is to use Speech-to-Text models to evaluate Text-to-Speech models, something similar to what we did to calculate the R-Error-Rate previously. That can be an improvement, but no doubt more research in this field is needed to find ways and develop methodologies to better asses other linguistic features of generated speech, such as sparsity, intelligibility, duration, prosody, and so on. Closely related to this, the model presented here has speaker conversion. This has only been slightly tested with human evaluation, as it was not the focus of the research, but doing a more profound zero-shot speaker evaluation is another pending task for the future.

Last but not least, with respect to the Speech-to-Speech Translation prototype created, there is a lot of work pending, especially for the Speech-to-Text transcription module and the Machine Translation part. The STT systems used here are Deep Speech (Hannun et al., 2014a) models; these are thoroughly tested, lightweight architectures but slightly outdated models. It might be interesting to try more recent audio transcription models like QuartzNet (Kriman et al., 2019) or Conformers (Gulati et al., 2020), which, although they are somewhat more complicate to create and train, usually give better results. Related to this, it would be interesting to evaluate the S2ST model as a whole too. To evaluate translation speech models, there exist solutions to generate text-based metrics using intermediate ASR models, like ASR-SENTBLEU (Jia et al., 2019) and ASR-COMET (Rei et al., 2020). There also exist some recent models trained on human quality scores, but there is an evident shortage of real human metric datasets, so these models have a hard time doing well (Rei et al., 2020). Another alternative would be to use recent text-free metric evaluation models like BLASTER (Chen et al., 2022).

6.4 Contributions of the Thesis

During the research carried out here, multiple Text-to-Speech multilingual models have been developed and evaluated. As a final result, there is a successfully trained speech synthesizer supporting English, Basque, Spanish, Portuguese, French, Catalan, and Galician. This model can be used for future research, projects, and applications, which include a large number of tasks and sectors. Along the way, multiple packages related to dataset pre-processing and Speech-to-Text models have been shared with the team and the community, like the multilingual *normalize-text* pre-processing tool, the improved Basque STT model, and we have released the first Deep Speech model trained on the Galician language. At the same time, the full process has been documented here for others to replicate, improve, or as a source of inspiration and ideas for other similar projects and research.

Additionally, a work-in-progress Speech-to-Speech Translation project has been started that will continue to be developed; new, better models will be added as the state-of-the-art advances and becomes more accessible, and new datasets will be created and shared.

6.5 Summary and Conclusion

In this research, a new multilingual Text-to-Speech has been successfully developed, which includes Basque, Spanish, and English, among others. The full training process included dataset pre-processing, consequent model training in multiple steps, and an evaluation as thoughtful as the available resources allowed, with everything documented along the way. The obtained results suggest that not only does the model learn new languages, but it also improves the quality of the previous models without showing any sign of degradation. As it is expected, the final deliverable produced here can be reused for other projects or continue being improved in the near future.

Generally speaking, taking into account the pace of change that the field of deep learning is taking, there is potential growth in this and the coming years that will affect speech-related models, including but not limited to speech synthesizers. The experiments carried out here to generate a state-of-the-art model that knows how to synthesize minority languages can constitute a good starting point to get on the bandwagon of neural model development in the field of speech. Additionally, this may help other medium-small size teams to contribute with other models or carry out further research, covering more languages and cultures.

On the journey to create these models, opportunities have arisen to launch other projects related to the field of speech and natural language processing, such as Speech-to-Text, Machine Translation, and Speech-to-Speech Translation models.

References

Rodrigo Agerri, Josu Bermudez, and German Rigau. IXA pipeline: Efficient and ready to use multilingual NLP tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3823–3828, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/775_Paper.pdf.

Rodrigo Agerri, Xavier Gómez Guinovart, German Rigau, and Miguel Anxo Solla Portela. Developing new linguistic resources and tools for the Galician language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1367.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019. URL https://arxiv.org/abs/1907.10902.

Jay Alammar. The illustrated transformer, 2018. URL http://jalammar.github.io/illustrated-transformer/.

Jonathan Allen, Sharon Hunnicutt, Rolf Carlson, and Bjorn Granstrom. Mitalk-79: The 1979 mit text-to-speech system. *The Journal of the Acoustical Society of America*, 65 (S1):S130–S130, 1979. doi: 10.1121/1.2017051. URL https://doi.org/10.1121/1.2017051.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus, 2019. URL https://arxiv.org/abs/1912.06670.

Sercan O. Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech, 2017. URL https://arxiv.org/abs/1702.07825.

Karunesh Arora, Sunita Arora, and Mukund Kumar Roy. Speech to speech translation: a communication boon. *CSI Transactions on ICT*, 1(3):207–213, Sep 2013. ISSN 2277-9086. doi: 10.1007/s40012-013-0014-4. URL https://doi.org/10.1007/s40012-013-0014-4.

Mikel Artetxe, Itziar Aldabe, Rodrigo Agerri, Olatz Perez-de Viñaspre, and Aitor Soroa. Does corpus quality really matter for low-resource languages?, 2022. URL https://arxiv.org/abs/2203.08111.

Vincent Aubanel, Maria Luisa García Lecumberri, and Martin Cooke. The sharvard corpus: A phonemically-balanced spanish sentence resource for audiology. *International Journal of Audiology*, 53(9):633–638, 2014. doi: 10.3109/14992027.2014.907507. URL https://doi.org/10.3109/14992027.2014.907507. PMID: 24863133.

Alan Black, Paul Taylor, Richard Caley, Rob Clark, Korin Richmond, Simon King, Volker Strom, and Heiga Zen. The festival speech synthesis system version 1.4.2. Software, Jun 2001. URL http://www.cstr.ed.ac.uk/projects/festival/.

A. Bonafonte, I. Esquerra, L. Aguilar, S. Oller, and M. Moreno. *Recent work on the FESTCAT database for speech synthesis*, chapter 1, pages 131–132. SIG-IL/Microsoft, 2009. URL http://hdl.handle.net/2117/9224.

John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf.

Edresson Casanova, Christopher Shulby, Eren Gölge, Nicolas Michael Müller, Frederico Santos de Oliveira, Arnaldo Candido Junior, Anderson da Silva Soares, Sandra Maria Aluisio, and Moacir Antonelli Ponti. Sc-glowtts: an efficient zero-shot multi-speaker text-to-speech model, 2021. URL https://arxiv.org/abs/2104.05557.

Edresson Casanova, Arnaldo Candido Junior, Christopher Shulby, Frederico Santos de Oliveira, João Paulo Teixeira, Moacir Antonelli Ponti, and Sandra Aluísio. TTS-portuguese corpus: a corpus for speech synthesis in brazilian portuguese. *Language Resources and Evaluation*, 56(3):1043–1055, jan 2022a. doi: 10.1007/s10579-021-09570-4. URL https://doi.org/10.1007%2Fs10579-021-09570-4.

Edresson Casanova, Arnaldo Candido Junior, Christopher Shulby, Frederico Santos de Oliveira, João Paulo Teixeira, Moacir Antonelli Ponti, and Sandra Aluísio. Tts-portuguese corpus: a corpus for speech synthesis in brazilian portuguese. *Language Resources and Evaluation*, pages 1–13, 2022b.

Jianfei Chen, Cheng Lu, Biqi Chenli, Jun Zhu, and Tian Tian. VFlow: More expressive generative flows with variational data augmentation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1660–1669. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/chen20p.html.

Mingda Chen, Paul-Ambroise Duquenne, Pierre Andrews, Justine Kao, Alexandre Mourachko, Holger Schwenk, and Marta R. Costa-jussà. Blaser: A text-free speech-to-speech translation evaluation metric, 2022. URL https://arxiv.org/abs/2212.08486.

Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, and Tao Qin. MultiSpeech: Multi-Speaker Text to Speech with Transformer. In *Proc. Interspeech 2020*, pages 4024–4028, 2020b. doi: 10.21437/Interspeech.2020-3139.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL `https://arxiv.org/abs/1409.1259`.

Junichi Yamagishi Christophe Veaux and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit, 2017.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *Interspeech 2018*. ISCA, sep 2018. doi: 10.21437/interspeech.2018-1929. URL `https://doi.org/10.21437%2Finterspeech.2018-1929`.

Cecil H. Coker. A model of articulatory dynamics and control. *Proceedings of the IEEE*, 64:452–460, 1976.

Erica Cooper, Wen-Chin Huang, Tomoki Toda, and Junichi Yamagishi. Generalization ability of mos prediction networks, 2021. URL `https://arxiv.org/abs/2110.02635`.

B.C. Csaji. Approximation with artificial neural networks. *M.S.'Thesis, Dept. Science, Eotvos Lorand Univ., Budapest, Hungary*, 2001. URL `https://cir.nii.ac.jp/crid/1573105974142800512`.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Interspeech*, 2020.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014. URL `https://arxiv.org/abs/1410.8516`.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2016. URL `https://arxiv.org/abs/1605.08803`.

Homer Dudley and Thomas H. Tarnóczy. The speaking machine of wolfgang von kempelen. *Journal of the Acoustical Society of America*, 22:151–166, 1949.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows, 2019. URL `https://arxiv.org/abs/1906.04032`.

Gölge Eren and The Coqui TTS Team. Coqui TTS, 1 2021. URL `https://github.com/coqui-ai/TTS`.

D Erro, Inmaculada Hernáez, Agustin Alonso, D García-Lorenzo, Eva Navas, J Ye, H Arzelus, I Jauk, N Hy, Carmen Magariños, M Sulír, Xiaohai Tian, X Wang, and Ruben Perez Ramon. Personalized synthetic voices for speaking impaired: Website and app. In *Interspeech*, volume 2015, 09 2015.

Daniel Erro, Inma Hernáez, Eva Navas, Agustín Alonso, Haritz Arzelus, Igor Jauk, Nguyen Quy Hy, Carmen Magariños, Rubén Pérez-Ramón, and Jianpei Ye. Zuretts: Online platform for obtaining personalized synthetic voices. In *eNTERFACE'14*, 2014. URL https://aholab.ehu.eus/eNTERFACE14/reports/enterface14_report_zuretts.pdf.

Yuchen Fan, Yao Qian, Fenglong Xie, and Frank K. Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. In *Interspeech*, 2014.

Cedric Gegout, Bernard Girau, and Fabrice Rossi. A mathematical model for feed-forward neural networks : theoretical description and parallel applications. Research Report LIP RR-1995-23, Laboratoire de l'informatique du parallélisme, September 1995. URL https://hal-lara.archives-ouvertes.fr/hal-02101945.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Alexander Grossmann and Jean Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *Siam Journal on Mathematical Analysis*, 15:723–736, 1984.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020. URL https://arxiv.org/abs/2005.08100.

Murilo Gustineli. A survey on recently proposed activation functions for deep learning, 2022. URL https://arxiv.org/abs/2204.02921.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014a. URL https://arxiv.org/abs/1412.5567.

Awni Y. Hannun, Andrew L. Maas, Daniel Jurafsky, and Andrew Y. Ng. First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns, 2014b. URL https://arxiv.org/abs/1408.2873.

M. Hansen et al. Gruut: A tokenizer, text cleaner, and ipa phonemizer for several human languages that supports ssml. https://github.com/rhasspy/gruut, 2022.

Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W11-2123.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2121.

Hee Soo Heo, Bong-Jin Lee, Jaesung Huh, and Joon Son Chung. Clova baseline system for the voxceleb speaker recognition challenge 2020, 2020. URL https://arxiv.org/abs/2009.14153.

Inma Hernaez, Eva Navas, Juan Luis Murugarren, and Borja Etxebarria. Description of the AhoTTS system for the Basque language. In *Proc. 4th ISCA ITRW on Speech Synthesis (SSW 4)*, page paper 202, 2001.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2722–2730. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ho19a.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, jul 1989. ISSN 0893-6080.

Wen-Chin Huang, Erica Cooper, Yu Tsao, Hsin-Min Wang, Tomoki Toda, and Junichi Yamagishi. The voicemos challenge 2022, 2022. URL https://arxiv.org/abs/2203.11389.

Andrew J. Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1:373–376 vol. 1, 1996.

Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. Direct speech-to-speech translation with a sequence-to-sequence model, 2019. URL https://arxiv.org/abs/1904.06037.

Wolfgangs Von Kempelen. *Wolfgangs Von Kempelen, Mechanismus der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine*, pages 1–456. Wien: J.B. Degen, 1791. doi: https://doi.org/10.6083/sx61dm64r.

Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search, 2020. URL https://arxiv.org/abs/2005.11129.

Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech, 2021. URL https://arxiv.org/abs/2106.06103.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Oddur Kjartansson, Alexander Gutkin, Alena Butryna, Isin Demirsahin, and Clara Rivera. Open-Source High Quality Speech Datasets for Basque, Catalan and Galician. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 21–27, Marseille, France, May 2020. European Language Resources association (ELRA). ISBN 979-10-95546-35-1. URL https://www.aclweb.org/anthology/2020.sltu-1.3.

Dennis H. Klatt. Software for a cascade/parallel formant synthesizer. *The Journal of the Acoustical Society of America*, 67(3):971–995, 1980. doi: 10.1121/1.383940. URL https://doi.org/10.1121/1.383940.

Dennis H. Klatt. Review of text-to-speech conversion for english. *The Journal of the Acoustical Society of America*, 82(3):737–793, 1987. doi: 10.1121/1.395275. URL https://doi.org/10.1121/1.395275.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis, 2020. URL https://arxiv.org/abs/2010.05646.

Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions, 2019. URL https://arxiv.org/abs/1910.10261.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. Melgan:

Generative adversarial networks for conditional waveform synthesis, 2019. URL https://arxiv.org/abs/1910.06711.

Kevin LaGrandeur. The talking brass head as a symbol of dangerous knowledge in friar bacon and in alphonsus, king of aragon. *English Studies*, 80(5):408–422, 1999. doi: 10.1080/00138389908599194. URL https://doi.org/10.1080/00138389908599194.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2015. URL https://arxiv.org/abs/1512.09300.

Alon Lavie, Alex Waibel, Lori Levin, , Donna Gates, , Torsten Zeppenfeld, and Puming Zhan. Janus iii: Speech-to-speech translation in multiple languages. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, volume 1, pages 99 – 102, April 1997.

Gianni Lazzari. TC-STAR: a speech to speech translation project. In *Proceedings of the Third International Workshop on Spoken Language Translation: Plenaries*, Kyoto, Japan, November 27-28 2006. URL https://aclanthology.org/2006.iwslt-plenaries.1.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Neural speech synthesis with transformer network, 2018. URL https://arxiv.org/abs/1809.08895.

Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling, 2016. URL https://arxiv.org/abs/1609.01454.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks, 2016. URL https://arxiv.org/abs/1611.04076.

Ignatius G. Mattingly. *Speech Synthesis for Phonetic and Phonological Models*, pages 1–37. Mouton, The Hague, 1974. doi: https://doi.org/10.6083/sx61dm64r.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502, 2017. doi: 10.21437/Interspeech.2017-1386.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In *Interspeech 2021*. ISCA, aug 2021. doi: 10.21437/interspeech.2021-299. URL https://doi.org/10.21437%2Finterspeech.2021-299.

Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9(5):453–467, 1990. ISSN 0167-6393. doi: https://doi.org/10.1016/0167-6393(90)90021-Z. URL https://www.sciencedirect.com/science/article/pii/016763939090021Z. Neuropeech '89.

Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf.

J. Olive. Rule synthesis of speech from dyadic units. In *ICASSP '77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 568–570, 1977. doi: 10.1109/ICASSP.1977.1170350.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016a. URL https://arxiv.org/abs/1609.03499.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016b. URL https://arxiv.org/abs/1606.05328.

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA, sep 2019. doi: 10.21437/interspeech.2019-2680. URL https://doi.org/10.21437/interspeech.2019-2680.

Jon D. Paul. The origins of audio and video compression: Some pale gleams from the past. In *SMPTE 2014 Annual Technical Conference & Exhibition*, pages 1–18, 2014. doi: 10.5594/M001572.

Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao. Non-autoregressive neural text-to-speech, 2019. URL https://arxiv.org/abs/1905.08459.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning, 2017. URL https://arxiv.org/abs/1710.07654.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech, 2021. URL https://arxiv.org/abs/2105.06337.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis, 2018. URL `https://arxiv.org/abs/1811.00002`.

M. Winterbottom R. M. Thomson. *William of Malmesbury: Gesta Regum Anglorum, The History of the English Kings*, pages 1–912. Oxford University Press, 1999.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL `https://aclanthology.org/2020.emnlp-main.213`.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech, 2019. URL `https://arxiv.org/abs/1905.09263`.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech, 2020. URL `https://arxiv.org/abs/2006.04558`.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015. URL `https://arxiv.org/abs/1505.05770`.

Amarildo Rista and Arbana Kadriu. Automatic speech recognition: A comprehensive survey. *SEEU Review*, 15:86–112, 12 2020. doi: 10.2478/seeur-2020-0019.

Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

E Rodríguez Banga, C García-Mateo, F Méndez-Pazó, M González-González, and C Magarinos. Cotovía: an open source tts for galician and spanish. In *VII Jornadas en Tecnología del Habla and III Iberian SLTech Workshop, IberSPEECH*, 2012.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

D. E. Rumelhart, Geoffrey E. Hinton, and R William. Learning representations by back-propagation errors, nature. In *Nature*, page 533–536, 1986.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022, 2022. URL `https://arxiv.org/abs/2204.02152`.

Yoshinori Sagisaka, Nobuyoshi Kaiki, Naoto Iwahashi, and Katsuhiko Mimura. ATR μ-talk speech synthesis system. In *Proc. 2nd International Conference on Spoken Language Processing (ICSLP 1992)*, pages 483–486, 1992. doi: 10.21437/ICSLP.1992-125.

Iñaki Sainz, Daniel Erro, Eva Navas, Inma Hernáez, Jon Sanchez, Ibon Saratxaga, and Igor Odriozola. Versatile speech databases for high quality synthesis for basque. In *LREC*, pages 3308–3312. Citeseer, 2012.

Matti Karjalainen Sami Lemmetty. Review of speech synthesis technology. *Department of Signal Processing and Acoustics, Aalto University*, 1999.

Peter M. Seeviour, John N. Holmes, and Michael W. Judd. Automatic generation of control signals for a parallel formant speech synthesizer. In *ICASSP*, 1976.

Christine H. Shadle and Robert I. Damper. Prospects for articulatory synthesis: A position paper. In *Speech Synthesis Workshop*, 2001.

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. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2017. URL `https://arxiv.org/abs/1712.05884`.

Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306, mar 2020. doi: 10.1016/j.physd.2019.132306. URL `https://doi.org/10.1016%2Fj.physd.2019.132306`.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/sohl-dickstein15.html`.

I. Solak. The m-ailabs speech dataset. *Munich Artificial Intelligence Laboratories GmbH*, 2017. URL `url{https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/}`.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2020. URL `https://arxiv.org/abs/2011.13456`.

Branislav Sredojev, Dragan Samardzija, and Dragan Posarac. Webrtc technology overview and signaling solution design and implementation. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1006–1009, 2015. doi: 10.1109/MIPRO.2015.7160422.

Antti Santeri Suni, Daniel Aalto, Tuomo Raitio, Paavo Alku, and Martti Vainio. Wavelets for intonation modeling in hmm speech synthesis. In Antonio Bonafonte, editor, *8th ISCA Workshop on Speech Synthesis, Proceedings, Barcelona, August 31 - September 2, 2013*, pages 285–290, France, 2013. ISCA. 8th ISCA Speech Synthesis Workshop ; Conference date: 31-08-2013 Through 02-09-2013.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014. URL https://arxiv.org/abs/1409.3215.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis, 2021. URL https://arxiv.org/abs/2106.15561.

Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, Frank Soong, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Naturalspeech: End-to-end text to speech synthesis with human-level quality, 2022. URL https://arxiv.org/abs/2205.04421.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raul Vazquez, and Sami Virpioja. Democratizing machine translation with opus-mt, 2022. URL https://arxiv.org/abs/2212.01936.

Francis M. Tyers and Josh Meyer. What shall we do with an hour of data? speech recognition for the un- and under-served languages of common voice, 2021. URL https://arxiv.org/abs/2105.04674.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification, 2017. URL https://arxiv.org/abs/1710.10467.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis, 2017. URL https://arxiv.org/abs/1703.10135.

Lilian Weng. From gan to wgan. *lilianweng.github.io*, 2017. URL https://lilianweng.github.io/posts/2017-08-20-gan/.

Lilian Weng. From autoencoder to beta-vae. *lilianweng.github.io*, 2018a. URL https://lilianweng.github.io/posts/2018-08-12-vae/.

Lilian Weng. Flow-based deep generative models. *lilianweng.github.io*, 2018b. URL https://lilianweng.github.io/posts/2018-10-13-flow-models/.

Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021. URL `https://lilianweng.github.io/posts/2021-07-11-diffusion-models/`.

Frank. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:196–202, 1945.

Detai Xin, Yuki Saito, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari. Cross-Lingual Speaker Adaptation Using Domain Adaptation and Speaker Consistency Loss for Text-To-Speech Synthesis. In *Proc. Interspeech 2021*, pages 1614–1618, 2021. doi: 10.21437/Interspeech.2021-897.

Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation, 2020. URL `https://arxiv.org/abs/2002.07526`.

Takayoshi Yoshimura. Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems. *PhD diss, Nagoya Institute of Technology*, 2002.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. *6th European Conference on Speech Communication and Technology (Eurospeech 1999)*, 1999.

Heiga Zen, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7962–7966, 2013.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech, 2019. URL `https://arxiv.org/abs/1904.02882`.