Information Systems and
Machine Learning Lab.
Stiftung Universität Hildesheim
Marienburger Platz 22
31141 Hildesheim
Prof. Dr. Dr. Lars Schmidt-Thieme
Shayan Jawed

Student Research Project 2019/2020

Speech Synthesis and Translation via Exemplar Autoencoders

Ajith Gumudavelly, 305643

Can Shenol Berk, 306919

Sadiki Tyty Christian, 303397

Haruki Honda, 305304

Table of Contents

Abstract

In this report, we presented a way to make our target speaker utter any translated content in a language where one has never spoken while maintaining the target speaker's voice and appropriate accent for the language. Our final models consist of a combination of translation and audio synthesis parts. There are various models and papers regarding the translation part and audio synthesis part separately, however, the combination has not been investigated yet. Therefore, we investigated translation and audio synthesis parts separately and combined them at the end in this new approach. In the translation part, we investigated translation frameworks with pretrained models from English to French support and mainly benefited from the EPSnet translation toolkit. Speech input is an English speech of any individual to be translated to French speech. We used the cascaded model of end-to-end speech translation and text to speech models here. In the audio synthesis part, we followed an unsupervised approach using the exemplar autoencoder to maintain the originality of targeted speakers' style and features. Challenging part of this task is to generate the voice without missing one's style or features when they speak a translated content in an appropriate accent. To address this challenge, we used datasets from English speaking, non-French speakers, celebrity audios and conducted experiments with Obama and Oprah. End model results are evaluated with Mean Opinion Score (MOS) score in terms of naturalness, voice similarity and content consistency. Our model achieved good results in voice similarity compared to the benchmark models. They can be used in different streams from Movies to News such as recreating old documentaries with a voice in the background in different languages.

Chapter 1: Introduction

Language translations have a huge impact in major fields in the world, from movies to politics. An abundance of papers came into existence working on translation with different topics. We aim to show that one can speak a language in an appropriate accent without knowing the language. In the literature there is not available work accomplishing this. Therefore, we wanted to tackle this challenging task. This system would be beneficial in many industries, especially in the entertainment industry such as games, bilingual movies across the world, etc.

This report is written for the Student Research Project (SRP) 2019-2020. SRP is a yearly project-based course which is concerned with four to six student groups doing research under the supervision on the topics like Reinforcement Learning, Time-Series, etc. We are a group of four students and showing our research in this report along with the github link[1] regarding our implementation.

Normally the speech translation refers to conversion of speech from one language to another language which is a traditional way and already done by some frameworks or systems like Google Translation. Although these systems are generally accepted, they still have a long way to go as far as voice conversion is concerned. Therefore, we have considered voice conversion as our main entity and attempted to produce a translated audio with different targeted speakers' voices. The challenging part is to carry the target speaker's voice characteristics to the translated speech while maintaining naturalness of the target language.

We tackled this problem by using deep neural networks whereby the Audio Synthesis part consists of learning an autoencoder of the target speaker to capture their voice features for the voice conversion task. In this process, the encoder takes speech signals and captures characteristics of a speaker and then the decoder part takes the output of the encoder and predicts a target sequence as Mel spectrograms. Finally, there is a vocoder to generate audio from Mel spectrograms of decoder output.

At the beginning, we tried a more straightforward approach in translation by direct speech to speech translation. Direct speech-to-speech translation has always fascinated researchers in translation. This approach eliminated intermediary text to be translated in the process, however, there were obstacles for us to implement this. Therefore, we switched our plan to using a more common translation approach with intermediary components such as Automatic Speech Recognition (ASR), Machine Translation (MT) and Text-to-Speech synthesis (TTS). The majority of translation systems developed so far have heavily relied on these intermediate components. Our second approach was using all cascaded models of the above mentioned components. At the end, we end up with End-to-end Speech Translation which is translation of source language speech to target language text and cascading it to Text-to-Speech (TTS) model. Translation process will be elaborated in the methodology section.

[1] https://github.com/Student-Research-Project/Speech-Synthesis-and-Translation-via-Exemplar-Autoencoders

Figure 1 demonstrates our unsupervised approach to converting English speech input into French speech while maintaining the target speaker's stylistic prosody. The upper part of the figure shows the translation process in general and below part shows the audio synthesis part in a similar manner.

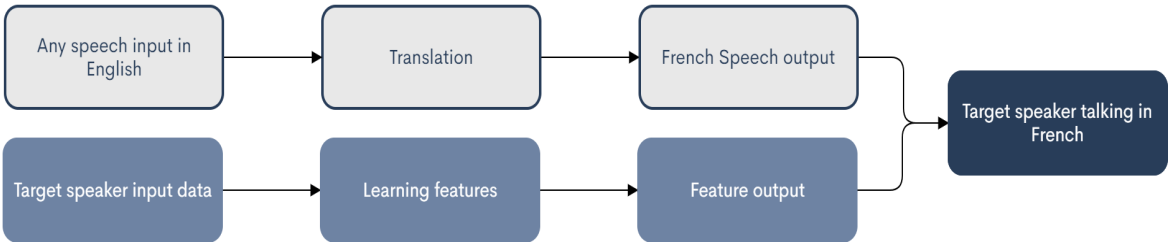

Figure 1: Model Architecture Overview

In the translation part, any speech in English can be a source input for the translation task. Using the translation framework and models, this content is translated into corresponding French speech as an output.

From the audio synthesis part, we obtained a target speaker input data . For example, it might be audio files of the speaker where this person speaks English in an easily audible way. From this, the model learns the features of the target speaker using the speech signals and mel spectrograms. Such that we have a feature output.

As a result, the model produces an French speech output of a target speaker while maintaining one's own characteristics. These two parts will be elaborated in the methodology part in detail. The figure below explains the main idea of the project having Obama as our target speaker.

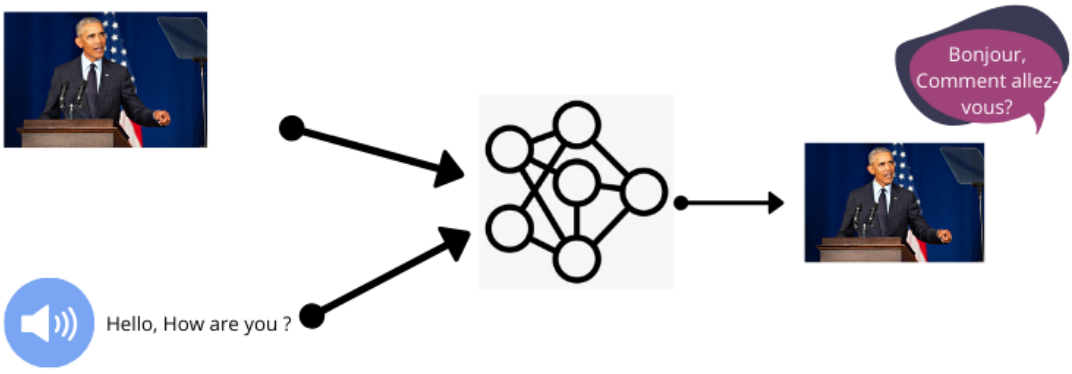

Figure 2: Process Example (Simplified)

It should be noted that any speech content can be source input as long as the target speaker has speech data in that language in order to make the model learn his/her vocal nuances and specific style. Even though our work and experiments are dependent on English to French translation, it can be adaptable to other languages in this way.

This approach addresses certain research questions: can we generate a clear synthetic voice? The output should be similar just like when the speaker speaks native language. The second question is: can the content of the translation be equally good? Translated speech content should keep the quality of the input speech along with the understandable accent. The third question is: can the translated synthetic speech sound natural? While the target speaker speaks in the language, it has to sound natural unlike robotic sound in which the quality is not upto the mark.

We used BLEU score for evaluating translated contents from English to French and Mean Opinion Score (MOS) for evaluating audio synthesis quality in terms of several criterias like naturalness in a generated audio. Our results are compared to other speech synthesis models in both English to English and English to French generated audios. We have good results in English to English speech generation and promising results with translation.

After the introduction, we are going to investigate background on the list of important papers in the development of translation and audio synthesis tasks. Then baseline papers will be elaborated and related works are going to be investigated in terms of comparable and benefited works in the next section. It will be followed by the explanation of methodology in translation part and audio synthesis part. These sections cover materials we used and the way we implemented as well as our different approaches to handle these tasks. After that, experiments and results of our findings will be elaborated. We used Obama and Oprah datasets for experiments and compared results with some benchmarks using MOS. Then, the discussion section includes future works and limitations. Finally, we provide a summary of our research.

1.1. Background

Here we have a list of papers which lead to the development of Machine Translation and Audio Synthesis: The first commercial machine translation systems were Rules-Based Machine Translation (RBMT) systems, which are based on linguistic rules that allow words to be located in different positions and have different meanings depending on the context. The RBMT technology is applied to broad sets of linguistic rules in three stages: study, transition, and creation. To its progress we have Example-Based Machine Translation (EBMT) which translates based on the comparison between bilingual corpus texts.

After the 1980s the computers started to develop rapidly. During the time we had some certain new machine translations such as statistical machine translation (SMT) and neural machine translations (NMT). SMT knows how to translate by statistically training on the bilingual data. In comparison to the word-based Rules-Based Machine Translation (RBMT) approach, most modern SMT systems are phrase-based and assemble translations using overlap phrases and

NMT is a method that employs a massive artificial neural network (ANN) to predict the probability of a series of terms, often in the form of entire sentences. Unlike statistical machine translation, which takes more memory and time to learn, neural machine translation trains an end to end model to optimize performance.

For audio synthesis we have Complex Audio Synthesis (CAS) during the 1970's which utilizes feedforward ANN rather than discriminative or regression tasks. In this scheme, an ANN is trained on low-level function frames. An auto encoding neural net is used to learn a high level representation of musical audio and during 2002 we have Virtual Audio Synthesis (VAS) where we synthesize the audio that may have been heard somewhere in the line linking the two microphones. The process operates in anechoic settings. There is no need to quantify the amount of sources present in the atmosphere or isolate the sources from the obtained audio mixtures to measure the interpolated audio. Regarding Real Time Audio Synthesis (RTAS) we have a paper which is based on bitwise logical modulation. This is an implementation that demonstrates the ability of basic elementary bitwise logical operations (OR, AND, XOR) to generate such spectra. When these operations were extended to two sinusoidal audio oscillators, they provided a wide range of new harmonically based sonic content. The last one is Musical Audio Synthesis which is quite interesting. This deals with real time musicians for music synthesis. There were a vast amount of developments happening both in machine translations and audio synthesis parts from the inception of technology.

Below are the list of the history of papers for both audio synthesis and machine translations in chronological order. In Machine Translation we have Rule based Machine Translation (1940-1970), Example based Machine Translation (1984), Statistical Machine Translation (1980-1990) and Neural Machine Translations (2014-2017). For audio synthesis we have real time Complex Audio Synthesis (1976-1998), Virtual Audio Synthesis (2002), Real Time Audio Synthesis (2004-2019) and Musical Audio Synthesis (2020).

Figure 3: History of Research Papers for Machine Translations

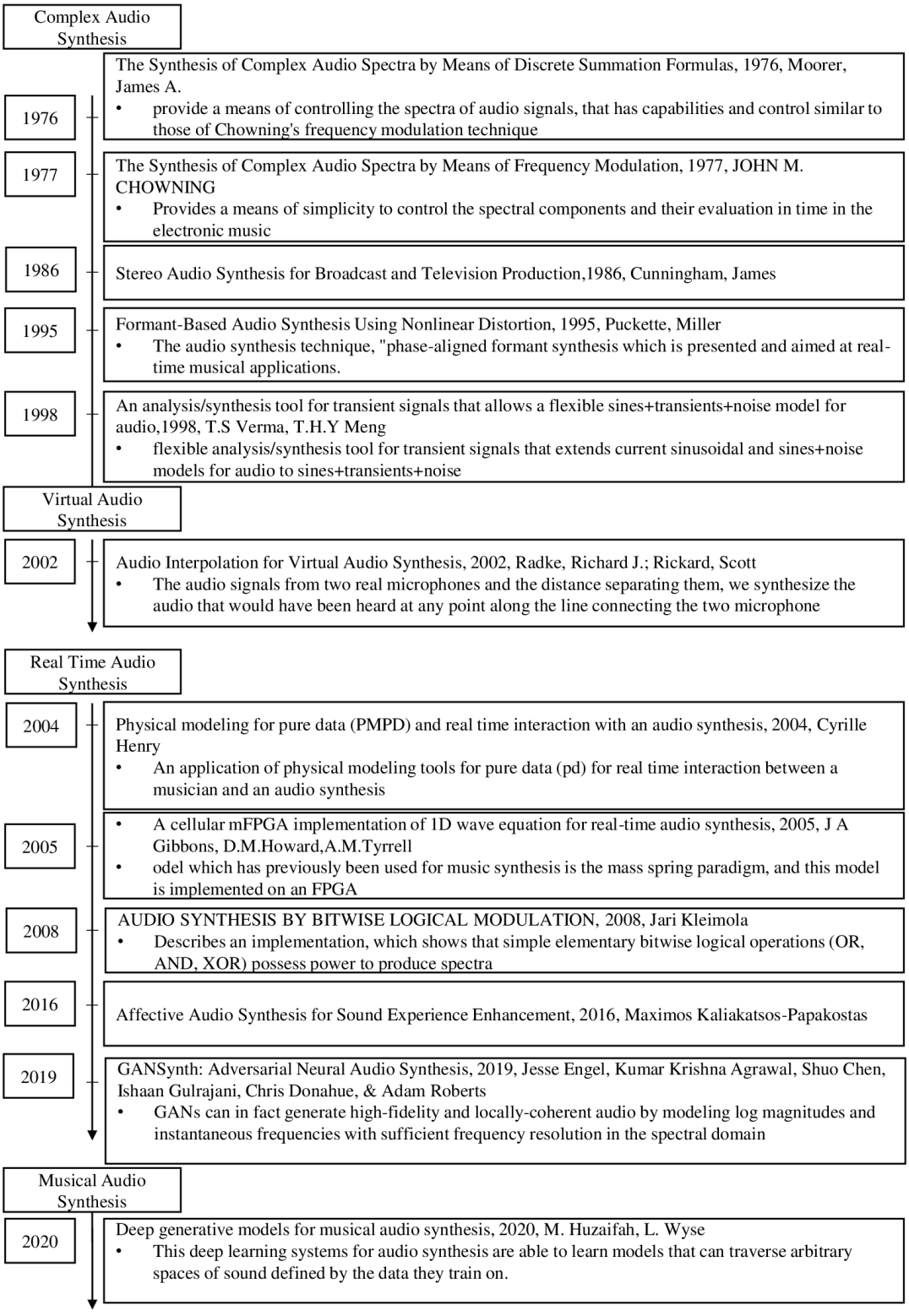

Figure 4: History of Research Papers for Audio Synthesis

Chapter 2: Baseline Papers

This chapter describes baseline papers for our research in both translation and audio synthesis aspects separately.

In the translation part, our focus is to obtain French speech output from English speech input. According to this, we have two baseline papers.

Firstly, ESPnet-ST: All-in-One Speech Translation Toolkit [1] written in 2020 by Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Enrique Yalta Soplin, Tomoki Hayashi and Shinji Watanabe. This paper presents the ESPnet framework which is useful for the development of speech-to-speech translation systems. From this paper we take the approach of translation of speech and it integrates and implements the systems like ASR, MT, ST and TTS functions. Authors provided various models for translation tasks where we could work with end to end ST or cascaded models. Also, translation results are significantly good even compared to some of the state-of-the-art performances.

Secondly, One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech [2] written in 2020 by Tomáš Nekvinda, Ondřej Dušek. This paper has an approach of Tacotron 2 which is multilingual speech synthesis and uses the parameter generation and tries to have the naturalness of multilingual speech by using more languages. Authors emphasized on the voice cloning and improved it by adversarial speaker classifier. This paper provides pretrained vocoder with English to French option which makes it more significant to us.

In the audio synthesis part, we have a baseline paper concerned with learning the style of a target speaker for voice conversion tasks.

The paper named Unsupervised Any-to-Many Audiovisual Synthesis via Exemplar Autoencoders [3] is written in 2020 by Kangle Deng, Aayush Bansal, Deva Ramanan. In this paper, the model builds on autoencoders in an unsupervised setting which projects out-of-sample data to the distribution of the training set which is motivated by Principal Component Analysis (PCA) autoencoders in order to learn the features of a given speaker. Also, environmental acoustics are emphasized and attempted to be captured. This model can convert any input speech to an output set of potentially-infinitely many speakers. Furthermore, authors also exhibited the usefulness of their approach for generating video from audio signals.

2.1. Related Works

Related works for the translation part and audio synthesis part are going to be described separately in a similar structure to the baseline section. Here, we have papers from comparable and benefited models as related works according to our research objective.

Plenty of research has been done on audio translations. There are comparable works in the translation part to ESPnet as translation frameworks. These models in general consist of some of the ASR, MT, TTS models and some of them are cascaded while others are end-to-end(E2E).
First of all, the problem of not relying on intermediate text representation and to translate from one language to another language is investigated by Direct speech-to-speech translation with a sequence-to-sequence model [4] paper written by Ye Jia et al. in 2019. The whole network is trained end-to-end, learning to map speech spectrograms into target spectrograms in another language, corresponding to the translated content with the change in voice.

Secondly, a paper named Lingvo: a Modular and Scalable Framework for Sequence-to-Sequence Modeling [5] written in 2019 by authors named Jonathan Shen, Patrick Nguyen, Yonghui Wu et al. introduces a new model called Lingvo developed by Google. This paper has a solution for sequence-to-sequence models under deep learning research. The lingvo models are useful as they are flexible to use, extensible and also easily customizable. It is closely related to ESPnet as it also offers various translation tasks both under cascaded and E2E options.

Third paper we have is OpenSeq2Seq: Extensible Toolkit for Distributed and Mixed Precision Training of Sequence-to-Sequence Models [6]. This paper is also used for sequence-to-sequence models which is based on OpenSeq2Seq, this is a toolkit which allows to explore the field of sequence-to-sequence architecture. This also creates a building blocks for training the encoder and decoder for Machine Translation and Automatic Speech recognition as our model also uses both of these systems.

Fourth of all, RETURNN as a Generic Flexible Neural Toolkit with Application to Translation and Speech Recognition [7] written in 2018 by Albert Zeyer, Tamer Alkhouli, Hermann Ney. This paper uses one of the new systems called RETURNN which is used for translations and speech recognition. The authors compare the decoding and fast training speed of RETURNN of attention models and Tensorflow beam search decoder. They also show that a layer wise pretraining scheme for recurrent attention models gives over 1% BLEU improvement absolute and it allows training the deeper recurrent encoder networks.

Fifth of all, Open Source Toolkit for Speech to Text Translation [8] published by Zenkel et al. in 2018. It is a framework based on Speech translation. In this paper the authors introduced an open source toolkit for it supported by both cascaded and E2E options, however, there is no TTS. This is one of the easy ways for the text translation. This provides a docker container, which has a pipeline for a few systems such as speech recognition system, sentence segmentation system and also attention-based translation system.

Then there are three papers introducing their translation toolkits and focused on ASR and MT tasks. First, FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling [9] written by the authors Myle Ott et al. in 2019. Authors emphasize on their user friendly, fast and easily extensible approach through interface between its models etc. Second, there is Tensor2Tensor for Neural Machine Translation [10] written in 2018 by Ashish Vaswani, Samy Bengio, Eugene Brevdo.The authors here concentrated on deep learning models and it has Tensor2Tensor which

is a deep learning model including the implementation of state of the art transformed models. It uses a self attention mechanism to improve its performance. Third, OpenNMT: Open-Source Toolkit for Neural Machine Translation [11] written in 2017 by Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. They claim the importance of efficiency and extensibility by making their work as a community-built toolkit. This one has impact on many other works in neural machine translation and it is extended with an ASR module later apart from its MT focus.

Also, there are two papers focused on the ASR module. First one is The Kaldi Speech Recognition Toolkit [12] written in 2011 by Daniel Povey et al. Many papers benefit from the Kaldi complete recipes to make speech recognition systems in their work such as ESPnet framework. It supports feature extraction, acoustic modelling, phonetic decision trees and language modelling. Second, Wav2letter++: The Fastest Open-source Speech Recognition System [13] written 2019 by Vineel Pratap et al. In This paper introduces wav2letter++ as fastest open source framework depending on deep learning. It uses the ArrayFire tensor library which is used for efficiency. The authors also explain the architecture and design of the wav2letter++ system and when comparing it to other major open-source speech recognition systems. In few cases wav2letter++ is more than faster than other optimized frameworks for training the end-to-end neural networks for speech recognition.

There are several papers related to models used in the TTS and audio synthesis part categorized as transfer learning, knowledge sharing, voice cloning, code switching and voice conversion.

Transfer learning is a method to reuse a model in a new task with its details. It is beneficial in using pre-trained model parameters to initialize new models. First paper is End-to-end Text-to-speech for Low-resource Languages by Cross-Lingual Transfer Learning [14] written in 2019 by Tao Tu, Yuan-Jui Chen, Cheng-chieh Yeh, Hung-yi Lee. Authors' goal is to construct a TTS system even with a low amount of data in target language by applying transfer learning from high source languages. They show that pronunciation knowledge can be maintained in the transferring procedure due to this studied mapping. Their method of learned mapping in the transfer learning process is beneficial to us. Second, Learning pronunciation from a foreign language in speech synthesis networks [15] written by Younggun Lee, Suwon Shon, Taesu Kim. This paper learns the new language using speech synthesis which learns the pronunciation from datasets from different languages using transfer learning from low resource language to high ones in TTS. Third of all, Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis [16] written by Ye Jia et al. in 2019.

Knowledge sharing is benefited in TTS part by joint training from multilingual data in this module. Paper named Building Multilingual End-to-End Speech Synthesisers for Indian Languages [17] written in 2019 by Anusha Prakash, Anju Leela Thomas, S. Umesh, Hema A Murthy. This paper builds TTS which is trained for the indian languages using two text representations character based, phone-based. As India has plenty of languages which makes it possible to apply the knowledge sharing technique.

There are various models to accomplish voice cloning tasks for the TTS part. Firstly, Learning to

Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning [18] written in 2019 by Yu Zhang, Ron J. Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, RJ Skerry-Ryan, Ye Jia, Andrew Rosenberg, Bhuvana Ramabhadran. Authors use phonetic feature representations and adversarial loss term to generate high quality speech synthesis in their Tacotron based TTS system. Second of all, a paper named VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop [19] written in 2018 by Yaniv Taigman, Lior Wolf, Adam Polyak, Eliya Nachmani. They introduce a new method for TTS without the necessity of phonemes or linguistic features with a simpler network architecture based on shifting buffer working memory. Third, Cross-lingual Multi-speaker Text-to-speech Synthesis for Voice Cloning without Using Parallel Corpus for Unseen Speakers [20] written in 2019 by Zhaoyu Liu and Brian Mak. They provide TTS for cross-lingual multi speaker system based on speaker encoder, a Tacotron-based synthesizer and a WaveNet vocoder. Its speaker encoder is phoneme-based Tacotron 2 with a ResCNN.

Code switching part is concerned with voice cloning in cross lingual settings. End-to-end Code-switched TTS with Mix of Monolingual Recordings [21] written in 2019 by Yuewen Cao et al.They investigated the E2E code switching (CS) system for TTS using a modified Tacotron. They combined outputs to apply CS on it. Their architecture has achieved satisfactory results in mono-lingual TTS.

Also, Char2Wav: End-to-End Speech Synthesis [22] written in 2017 by Jose Sotelo et al. The authors present it as an end-to-end model for speech synthesis while emphasizing the importance of naturalness and clarity of the voice. Char2Wav comprises two parts: a reader and a neural vocoder. There is an encoder decoder model with attention as the reader. Then, a bi-directional recurrent neural network is used as an encoder where inputs are phonemes and decoder is a RNN with attention that generates acoustic features for vocoder.

The Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions [23] is written by Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly. Speech synthesis system Tacotron 2 is presented in this paper. Character embeddings are mapped to mel spectrograms which are given as input to vocoder in the next stage. Vocoder is a modified version of WaveNet in this work which generates raw audio from mel-spectrograms. This is one of the important TTS papers that we benefited.

There are several papers regarding the voice conversion task. Firstly, Voice conversion through vector quantization [24] paper written by Masanobu Abe, Satoshi Nakamura, Kiyohiro Shikano, Hisao Kuwabara. In this paper, authors present a spectrum mapping method with vector quantization technique where they map different speaker codebooks through spectrum . Secondly, Continuous probabilistic transform for voice conversion [25] written in 1998 by Yannis Stylianou ,Olivier Cappé and Eric Moulines. This work presents a way to modify input speaker voice through Gaussian Mixture Model (GMM) approach to make it sound by another speaker. Third of all, a paper named Voice conversion using Artificial Neural Networks [26] written in 2009 by Srinivas Desai et al. benefited from ANN mapping abilities of source speaker features to a target speaker.

There are several voice conversion models concerned with fine-tune all the learned generic base models for target speaker features. First of all, AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss [27] written by Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang and Mark Hasegawa-Johnson. This paper focused on voice conversions of different speakers based on many-to-many and zero-shot. The authors use deeplstyle algorithms such as Generative Adversarial Networks and Conditional Variational Autoencoder. They present Auto-VC using their designed bottleneck with autoencoder and use autoencoder loss which resulted in a performance increase compared to traditional voice conversion models. Secondly, a paper named StarGAN-VC2: Rethinking Conditional Methods for StarGAN-Based Voice Conversion [28] written in 2019 by Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, Nobukatsu Hojo. Authors approached VC with non-parallel data and they used a method based on modulation to transform acoustic features in a domain-specific. Thirdly, Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion written in 2019 by Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, Nobukatsu Hojo. This paper is written by the same authors of [29] and interested in the same approach with non-parallel data. Difference is StarGAN is a multi-task version of CycleGAN. Fourthly, High-quality nonparallel voice conversion based on a cycle-consistent adversarial network [30] written by Fuming Fang, Junichi Yamagishi, Isao Echizen and Jaime Lorenzo-Trueba. Here in this paper for nonparallel data-based VC preparation, authors suggest using a cycle-consistent adversarial network. CycleGAN is a generative adversarial network designed for unpaired image-to-image conversion. The proposed approach greatly outperformed a method based on the Merlin open source neural network speech synthesis system and a GAN-based parallel VC system in an inter-gender transfer subjective evaluation.

Chapter 3: Methodology

3.1. Translation

The aim of the translation part is to make our target speaker be able to speak languages s/he has not spoken before. We used English speech as input and obtained French speech as output. According to this, we choose English native speakers like Obama, Oprah who cannot speak French at all.

There are four main terms to be used in this model. First of all, Automatic Speech Recognition (ASR) generates text of input speech. Secondly, Machine Translation (MT) converts text in a language into its corresponding text in another language. Thirdly, Speech Translation (ST) is a combination of ASR and MT where an input speech is converted to its corresponding text in another language. Finally, Text to Speech Synthesis (TTS) converts output from MT into its speech state.

At the end, we used ESPnet: End-to-End Speech Processing Toolkit [1] for automatic speech recognition (ASR) and machine translation (MT) tasks in their end-to-end ST (E2E-ST) aspect. For the text to speech task, we did not use ESPnet for performance reasons and preferred to benefit the work done in One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech paper [2] which has French language support. Furthermore, it has a different vocoder than ESPnet in order to generate human-like speech.

First, we will present what models and papers are investigated for the translation task and discuss the reasons whether to choose them or not. After that, we will focus on the details of ASR, MT and TTS of the chosen translation model.

3.1.1. Translation - Model Architecture

There are three options for translation tasks that we considered at the beginning.

First of all, an end-to-end sequential model including direct speech to speech translation where one language is translated to another language without text intermediaries. Figure 5 shows basic direct STS translation architecture.

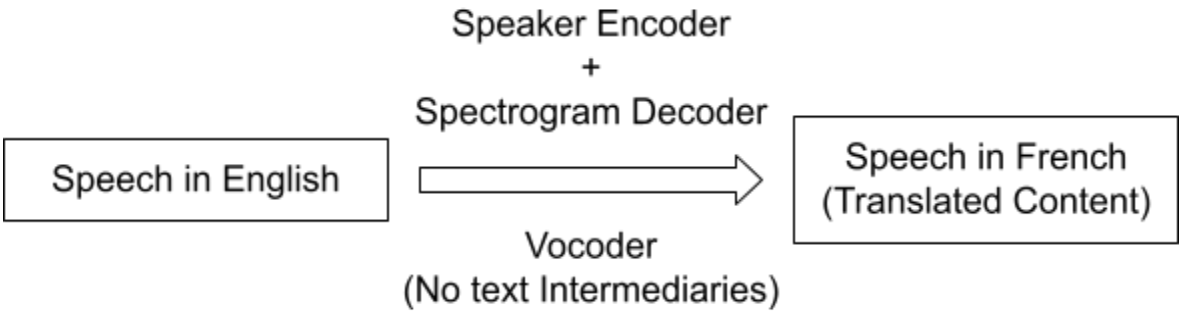

Figure 5: Direct STS overview

Secondly, a model with all cascaded parts can be an option. Figure 6 shows this approach as converting English speech input into its text in the ASR part, followed by using this text output in the MT model as input to obtain its corresponding French text translation and then generate French speech version of text obtained from previous MT task.

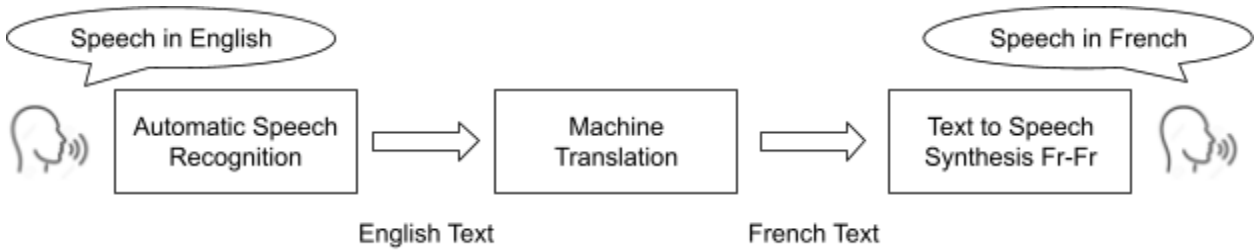

Figure 6: All Cascaded translation model overview

Finally, there is an end to end trained model of ASR and MT together which is called E2E-ST. Figure 7 shows that a speech input in source language (English) is directly mapped to its target language (French) translation in text which will be connected with TTS in the next stage.

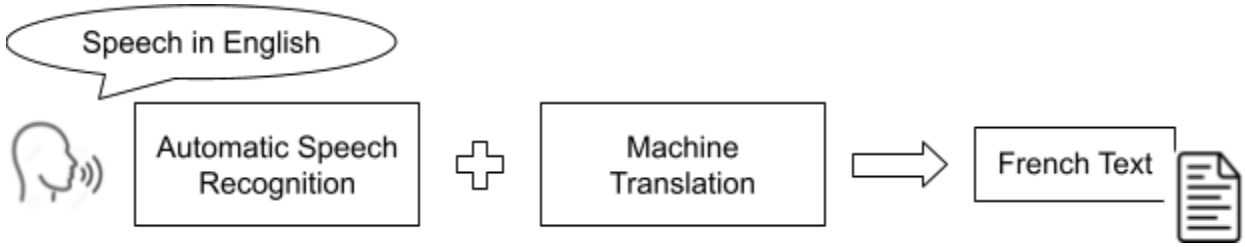

Figure 7: End to End Speech Translation (E2E-ST)

After the E2E-ST task, there is the TTS synthesis to generate speech output for target language (French) from the obtained text from the previous E2E-ST task. Figure 8 shows why this process is called cascaded in total because E2E-ST is followed by the TTS synthesis model.

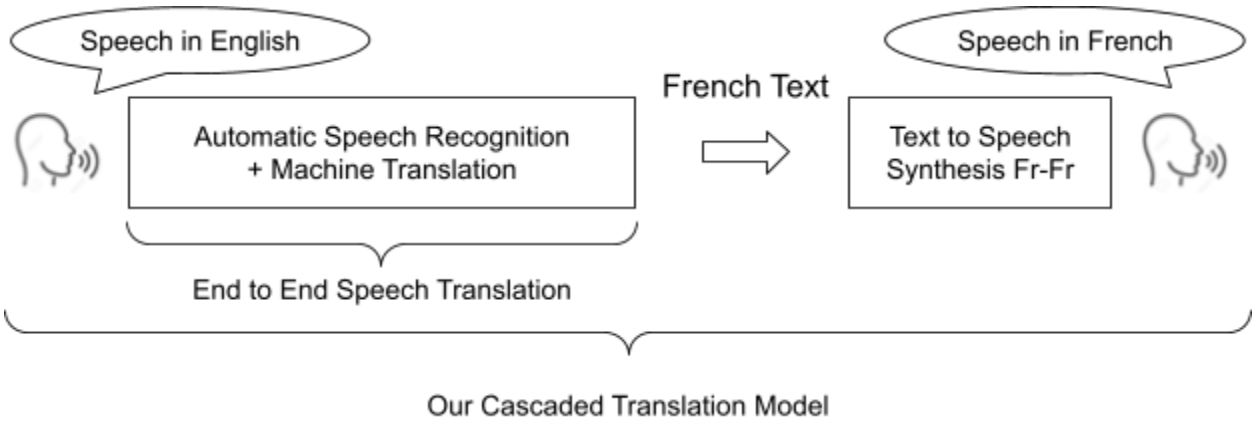

Figure 8: E2E-ST + TTS synthesis Model

Our initial idea was to have direct speech to speech translation [4]. It has the advantage of excluding the need to have text intermediaries and keeping the speaker voice. Even though we believe that this model would be more efficient to use, it is difficult to find material like source code or pretrained models because it is a recent developing area and also this model needs improvement in its results.

Result of [4] are underperforming its cascaded model baseline of S2T and TTS synthesis model. Moreover, the main reason why we did not stick to this model is insufficient resources we got and struggled to find more. Thus our initial plan has changed and we switched to using conventional speech to speech translation models. First, we investigated a model with all cascaded parts of ASR, MT and TTS synthesis, however, we did not consider it because we found that E2E-ST has advantages over cascaded ASR and MT such as decrease at inference time latency. We decided to look for a cascade model of E2E-ST and TTS as shown in Figure 7 and Figure 8. These components will be elaborated with corresponding parts of the chosen translation model.

3.1.2. Translation Framework Selection

It requires a lot of effort to make a translation toolkit which consists of thousands of lines of code. While paying our attention to the audio synthesis part, we wanted to obtain translation from a good and versatile platform with pretrained models.

Among conventional speech to speech translation sources we decided to use the ESPnet Translation toolkit which is an open source platform. Initially authors focused on the automatic speech recognition (ASR) part and published this toolkit for this area of utilization [33].

After revising their paper, authors make ESPnet adaptable toolkit by having End-to-End Speech Translation (E2E-ST) and Cascaded Speech Translation (Cascade-ST) with ASR, Language Modelling (LM), MT and TTS [1].

Even though authors believe that E2E-ST has several advantages over Cascade-ST, they present both approaches in their paper and point out that one does not perform significantly better than the other. Table 1 shows ESPnet tasks and also comparison among some of the translation toolkits for their supported tasks. We can see that ESPnet is the most comprehensive one among the compared toolkits.

Toolkit	Supported task						Example (w/ corpus pre-processing)						Pre-trained model
	ASR	LM	E2E-ST	Cascade-ST	MT	TTS	ASR	LM	E2E-ST	Cascade-ST	MT	TTS	
ESPnet-ST (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lingvo[1]	✓	✓	✓♣	✓♣	✓	✓♣	✓	✓	–	–	✓	–	–
OpenSeq2seq[2]	✓	✓	–	–	✓	✓	✓	✓	–	–	✓	–	✓
RETURNN[3]	✓	✓	✓	–	✓	–	–	–	–	–	–	–	✓
SLT.KIT[4]	✓	–	✓	✓	✓	–	✓	–	✓	✓	✓	–	✓
Fairseq[5]	✓	✓	–	–	✓	–	✓	✓	–	–	✓	–	✓
Tensor2Tensor[6]	✓	✓	–	–	✓	–	–	–	–	–	✓	–	✓◇
OpenNMT-{py, tf}[7]	✓	✓	–	–	✓	–	–	–	–	–	–	–	✓
Kaldi[8]	✓	✓	–	–	–	–	✓	✓	–	–	–	–	✓
Wav2letter++[9]	✓	✓	–	–	–	–	✓	✓	–	–	–	–	✓

Table 1: Toolkit overview for supported tasks [1,5,6,7,8,9,10,11,12,13]

3.1.3. ESPnet

We used ESPnet for the Translation task from English speech input to obtain French text output. There are three steps in this translation process. Two of them are explicitly shown in Figure 9 as Cascade-ST of ASR and MT models and also as E2E-ST. Last step is TTS synthesis and it will be covered after these steps. Although it follows a similar training flow consisting of six main stages to its former steps.

ESPnet resources are available both in Github and Google Colaboratory by authors. They provided Colab documents for ST and TTS parts. Looking at the Colab document, run.sh execution starts the recipes' main script. This process has 6 stages in total. Figure 10 shows that these stages follow Kaldi style for stages 0-2 and chainer/pytorch based DNN toolkit for training backend in stages 3-4. Also, they are similar in Cascade-ST and E2E-ST models by only difference of additional stage in the former one where evaluation depends on the feeding ASR output to the MT model (see stage 6 in Figure 9).

In this section, stages of E2E-ST will be investigated in detail. Also, TTS stages will be discussed briefly even though they are not used in the end model.

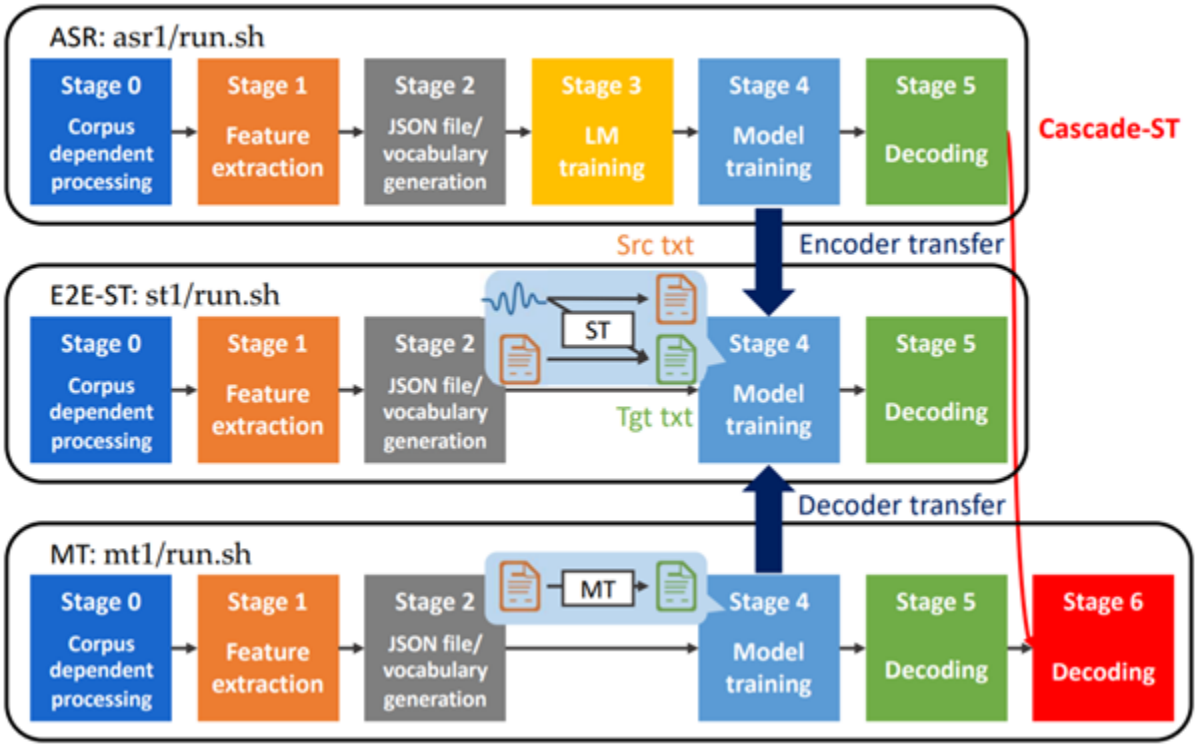

Figure 9: ESPnet Overview [1]

The Kaldi toolkit for ASR and WarpCTC are required to implement ESPnet and they are provided in the tools section by authors.

Figure 10 demonstrates the stages of E2E-ST concretely. Here are the stage details after executing the run.sh:

- Stage 0: Data should be prepared and Kaldi style data directories are created for training and evaluation sets.
- Stage 1: Feature extraction for speech is conducted in this stage with the help of Kaldi and they generated raw filterbanks. After that, ark and scp formats are used to save them as feature vector files for later statistics. They can be loaded to python through kaldiio tool. cmvn.ark is an important statistics part in this process as it helps to normalization of the features.

- Stage 2: Speech feature and transcription data are paired into JSON files. These JSON files consist of the shape, transcription, token and token id of its sequence. They are located under the /dump directory.
- Stage 3: Language model for ASR is trained. Millions of words text corpora used to estimate n-gram probabilities of word occurrences.
- Stage 4: ASR is trained with the help of Chainer and PyTorch backends. For the training configuration, .yaml format is used. Several network architectures are presented here for training configuration where best configuration can be found by tensorboard. Here are some examples:
    - RNN with attention+CTC (Connectionist Temporal Classification-loss function)
    - Transformer+CTC: Joint decoding with label-synchronous Joint CTC and transformer (Main approach)
    - RNN Transducer with attention consist of original RNN-T with attention
- Stage 5: Decoding and Scoring: In order to decode previously created yaml configuration, this formula is used as decoding score:

$$\text{argmax}_y (1-\lambda) \log P_{\text{dec}}(y \mid x) + \lambda \log P_{ctc}(y \mid x) + \gamma \log P_{lm}(y) + b|y| \quad \ldots\text{(eq. 1)}$$

Finally, sclite from SPTK toolkit is used for token error rate and word error rate to evaluate the results.

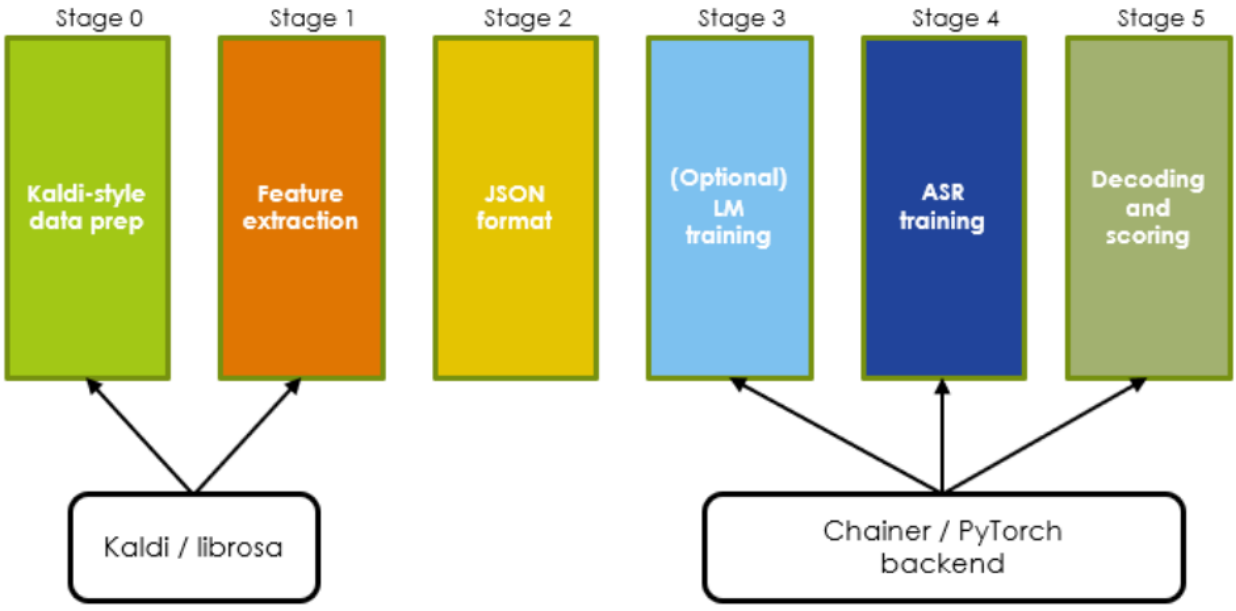

Figure 10: ESPnet stages

The TTS synthesis part follows the same approach in the first four stages to the ST part where it benefits from Kaldi style recipe to data preparation, feature extraction, normalization and text formatting. After that, training of TTS is exclusive to its configurations like Fastspeech even though it is written in a .yaml format file in a similar way. It supports three models Tacotron 2, Transformer and Fastspeech with their alternative variants. Further, a pre-trained WaveNet-vocoder is provided. It is possible to decode generated mel-spectrograms from the trained network and generate a waveform using Griffin-Lim algorithm. Also, it consists of four corpus such as English, Spanish, German and Japanese. However, it does not have French language support for TTS synthesis which we needed. That's why we got English-like accents from French content in ESPnet-TTS in the early stages of our work. Therefore, we changed this model by another one from [2]. This approach and implementation will be detailed in the TTS section.

So, let's look at ASR and MT model details followed by the E2E-ST model.

3.1.3.A. Automatic Speech Recognition

In the first part, there is the Automatic Speech Recognition Model, ESPnet uses open-source software called Kaldi toolkit which is used here for data processing and feature extraction. First, adapting kaldi, Figure 11 demonstrates input speech signal is processed to extract a series of features and then feature vector representation of that speech signal is denoted with X. They are 80-Dimensional log mel features with pitch features to be used in the model training.

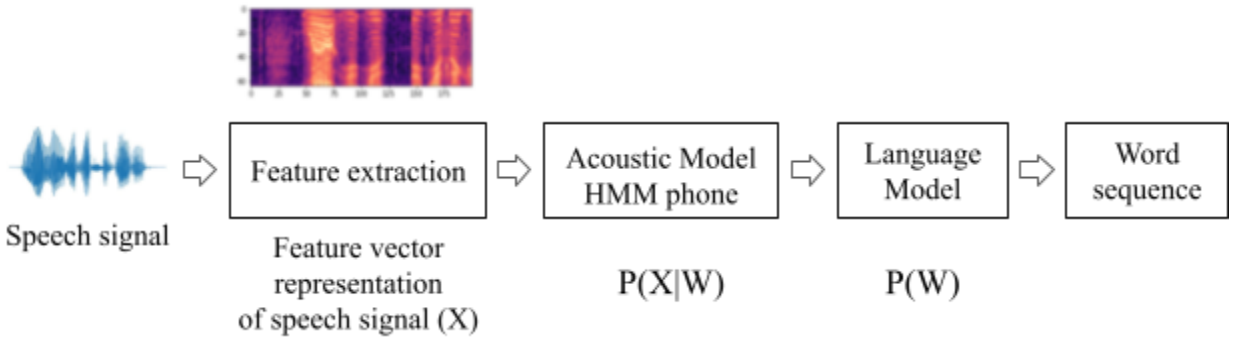

Figure 11: ASR Task

At the Automatic Speech Recognition training, Transformer based Hybrid CTC/attention encoder-decoder framework is trained by using the Chainer and PyTorch backend. Also, as a default attention it uses location-aware attention. Extracted features through kaldi are used to generate acoustic models through the likelihood of an observed acoustic signal X given a word sequence W. After that, in the language model, likelihood of an observed word sequence is derived by decoding the word sequence $\widehat{W}$ which maximizes equation in (2).

$$\begin{aligned}\hat{W} &= \arg\max_{W} P(W \mid X) \\ &= \arg\max_{W} \frac{P(W)P(X \mid W)}{P(X)} \\ &= \arg\max_{W} P(W)P(X \mid W)\end{aligned}$$

...(eq. 2)

3.1.3.B. Machine Translation

In the second stage, Machine Translation Model, there is a source text encoder for English text and translation decoder to obtain corresponding French text. Also, MT dataset has lower case source sentences without punctuation marks. Given word sequence in W in input language, most likely word sequence for target language $\widehat{A}$ can be computed as in equation (3). In this equation, P(W|A) represents the translation model and P(A) represents the target language model.

$$\begin{aligned}\hat{A} &= \arg\max_{A} P(A \mid W) \\ &= \arg\max_{A} \frac{P(A)P(W \mid A)}{P(W)} \\ &= \arg\max_{A} P(A)P(W \mid A)\end{aligned}$$

...(eq. 3)

Figure 12 demonstrates the process for text to text translation below. We already got the English text from the ASR model in the first stage and here we feed the Machine Translation model with output of the ASR.

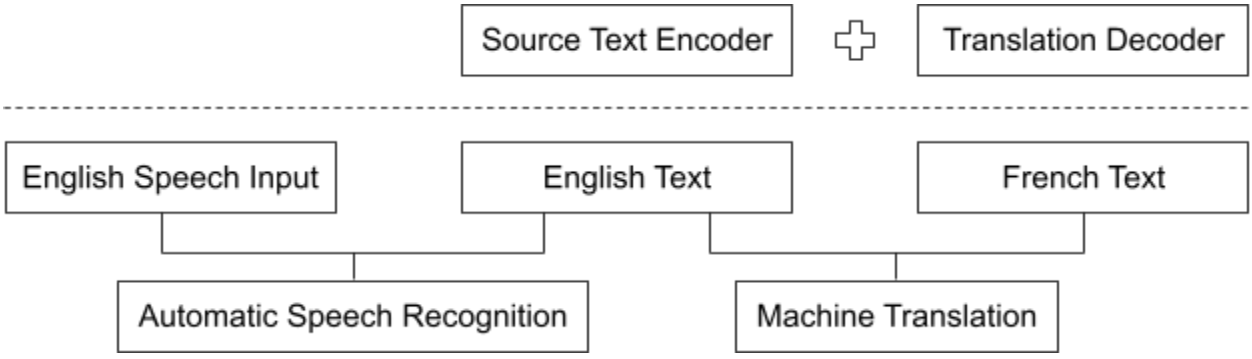

Figure 12: MT Task

However, we do not need the MT model explicitly as we used E2E-ST instead of Cascade-ST. It is good to have parameters from the MT model in order to use them in ST task for translation decoder initialization.

3.1.3.C. Speech Translation

Speech translation (ST) combines ASR and MT modules in a feasible computational way. Given input language feature vector X sequence, most likely word sequence $\widehat{A}$ can be computed. Input sentence W is a hidden variable, here in the equation (4), P(W|X) represents the ASR part and P(A|W) represents the MT part.

$$\begin{aligned}\widehat{A} &= \underset{A}{argmax}\ P(A|X) \\ &= \underset{A}{argmax}\ P(A, W|X) \\ &= \underset{A}{argmax} \sum_{W} P(A|W) * P(W|X) \\ &\cong \underset{A}{argmax} \left\{ \max_{W}\ P(A|W) * P(W|X) \right\}\end{aligned}$$

...(eq. 4)

E2E-ST has a speech encoder and translation decoder. Transfer learning approach allows to initialize ST encoder with ASR pre-trained parameters and ST decoder with MT pretrained parameters. It is possible to take parameters from pretrained models and use them in this combined task because parameter characteristics are exactly the same.

One of the performance problems is to overcome by having MT training data made out of source sentences with lowercase characters without punctuation to make it compatible with ASR output with similar sentence features.

In Figure 13, X sequence represents speech inputs and there is a sequence to sequence mapping function f(.) which stands for CTC, Attention based encoder decoder, Joint CTC/attention, RNN reducer and Transformer in ST case to obtain Y sequence of text. This demonstration can be adaptable to any ESPnet task as all of them are seq2seq problems by only changing the mapping function f(.) accordingly.

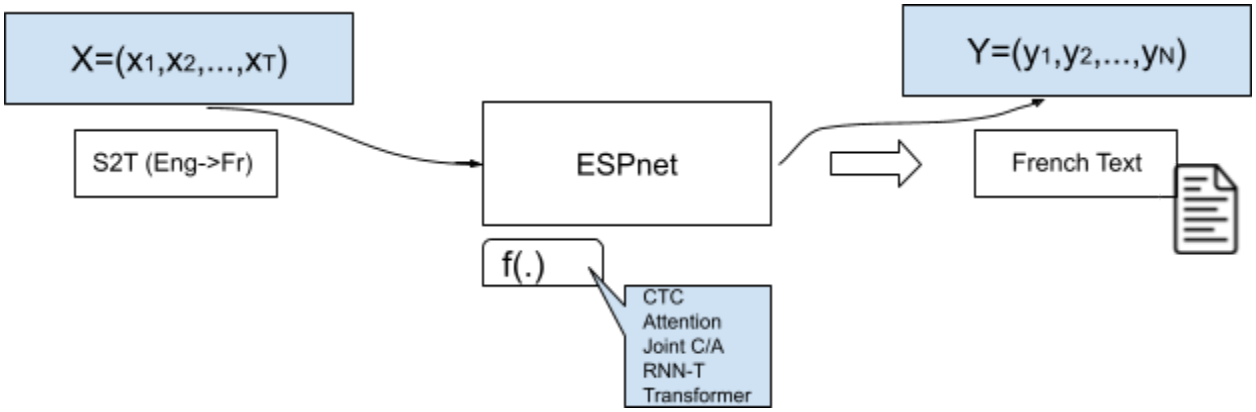

Figure 13: ESPnet Toolkit Design

Similar to Figure 11 in the ASR section, Figure 14 demonstrates an ordinary method for speech recognition to deal with sequence to sequence problems of translation. It consists of acoustic modelling, lexicon for pronunciation and language modelling.

Figure 14: Speech Recognition Pipeline

Speech recognition pipeline is integrated into End-to-End Neural Network architecture as demonstrated in Figure 15. This approach simplifies the process by integrating various models and this way the deep network is trained jointly to map an input signal to its target sequence in the target language directly. Thus, there is no need for intermediate training.

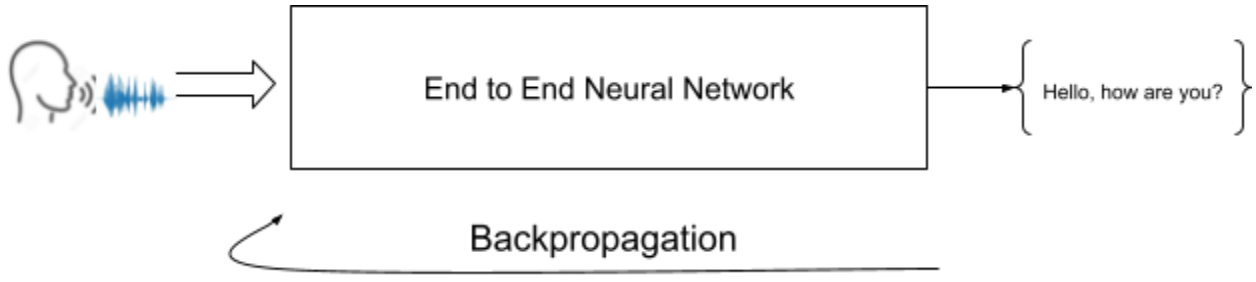

Figure 15: Integrated Architecture from the Pipeline

Figure 16 shows that other parts of speech processing such as the MT module can be integrated to this architecture in a similar way. Then multiple DNN's are combined and trained jointly. Moreover, the network can be optimized by backpropagation.

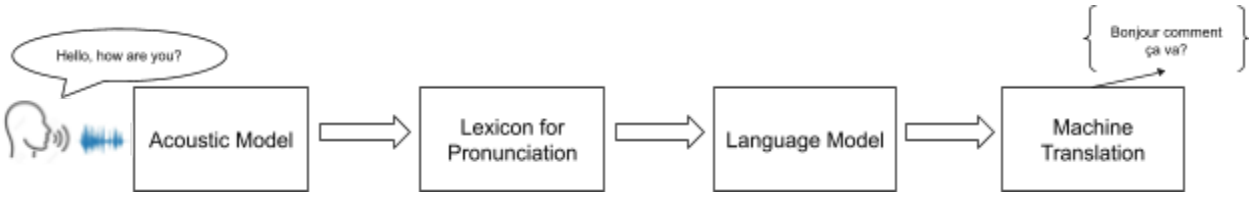

Figure 16: Integrating Other Modules to the Integrated Architecture

In the following step, we need a Text-to-Speech model for converting French text to French speech.

3.1.3.D. Text-to-Speech

ESPnet provides a text to speech model, however, there was certainly a problem of generating French audio synthesis because of English-like bad French accent. We looked for obtaining a better French accent in the final output and found a paper named One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech [2] which uses Tacotron 2 with WaveRNN based vocoder for the text to speech part and most importantly provide French language support. Thus, on top of the ESPnet translation model we used this model and obtained better results as shown in the Figure 17.

Figure 17: Combination of ESPnet [1] and TTS [2]

It is a well-known Neural Network architecture for speech synthesis directly from text. This system is composed of 2 main components. First, recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel spectrograms through its attention mechanism. Second component is WaveRNN based vocoder to generate raw audio.

Process in the first component is demonstrated in Figure 18 where the network consists of an encoder and decoder with an attention mechanism in between. Encoder converts text input sequence to hidden feature representation in 512-dimensional character embeddings. Then they are passed through 3 convolutional layers. After that attention mechanism uses encoder output to summarize a full encoded sequence as a fixed length context vector for each decoder output step. Finally, the decoder predicts mel-spectrograms from the encoded input sequence one frame at a

time. These mel-spectrograms are computed through Short-time-Fourier-Transform using 80 channel mel filter banks. They capture details of human speech like accent and voice.

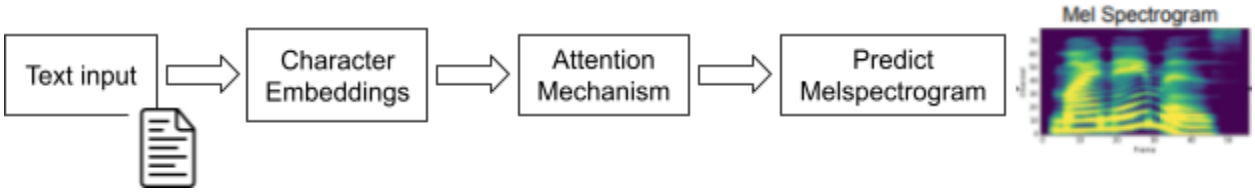

Figure 18: Mel Spectrogram Generation Process

In Figure 19, as input the model takes predicted 80-dimensional mel spectrogram frames from previous stage output and conditions on them to generate time-domain waveform samples which is basically an audio. Also, WaveRNN weights are pre-trained on CSS10 dataset.

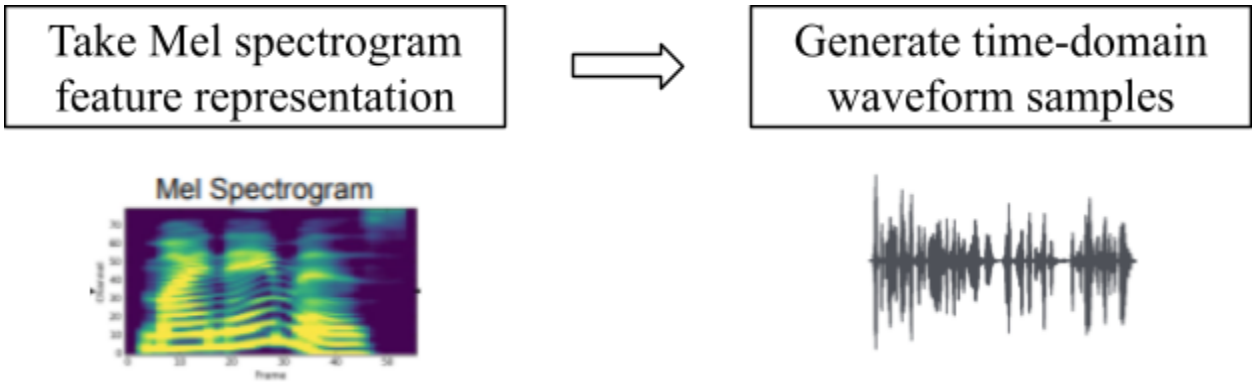

Figure 19: Audio Generation Process

3.2. Audio Synthesis

3.2.1. Voice Conversion (VC)

Voice conversion is a technique to modify the speech spoken by a source speaker so as to be perceived as that spoken by a specific target speaker. Several efficient approaches have been proposed, such as the codebook mapping method [24], the Gaussian mixture model (GMM) method [25], and the artificial neural network (ANN) method [26]. One obvious disadvantage of these methods is that the converted spectral features are limited to the discrete spaces, which will greatly degrade the perceptual quality of the converted speech. Hence, several improved approaches have been proposed.

The statistical methods have been prevalent in the past decades, and the GMM method is proven to be most popular and well-known for VC.

3.2.1.A. Traditional methods

Traditionally, the field of voice conversion and speech synthesis were dominated by statistical methods, especially Gaussian mixture models GMM for modeling the correspondence between source speech and the target speech features.

Back then, researchers in this field agreed that the vocal parameter of a speech and therefore any audio signal could be modeled as probabilistic distributions of some sort and most precisely as a Gaussian distribution.

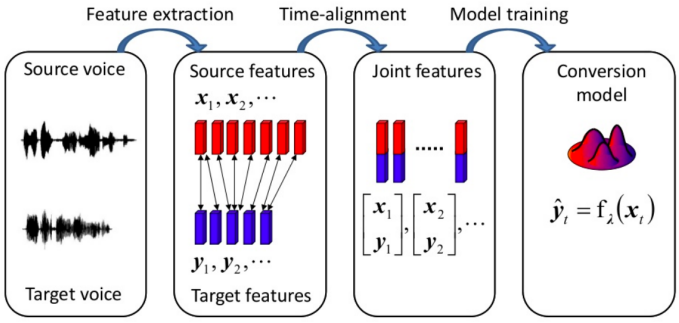

Figure 20: GMM models for voice conversion [25]

Spectral transformation using GMM models generally followed a basic template as shown in Figure 20 and further elaborated by the following points:

1) At the beginning, one needed to have a considerable amount of data points from both the source and target speaker, and in addition to that it had to be a parallel dataset. The conversion process began by aligning N spectral feature vectors in time (discrete cepstral coeffs). Here, it was important that source and target feature vectors are of the same lengths and shape to allow for alignment as shown by equation 5 below.

$$Source: X = \{x_{1,\ldots,}\, x_N\},\ target: Y = \left\{y_{1,\ldots,}\, y_N\right\},\ joint\ Z := (X, Y)$$

…(eq. 5)

2) Now, using basic learner such as expectation maximization (EM) algorithm one would represent the probability density function (PDF) of joint resultant vectors Z as a mixture or sum of Q multivariate Guassians distributed over the aligned source and target features in time domain.

$$p(z) = \sum_{q=1}^{Q} \alpha_q N\left(z; \mu_q, \sum_q\right),\ \sum_{q=1}^{Q} \alpha_q = 1,\ \alpha_q \geq 0$$

…(eq. 6)

$$Learn \left\{ \alpha_q, \mu_q, \sum_q, q = 1{:}Q \right\} from\ Expectation\ Maximization\ (EM)\ on\ Z$$

…(eq. 7)

3) And finally the output of the learned function is the probability of sound components mapped from the source to the target speaker. This is achieved by transforming source vectors using a weighted mixture of Maximum likelihood (ML) estimator for each component as shown in equation 8. However in reality this process is much more complicated than that and requires much more computation resources.

$$\hat{y}_n(X_n) = \sum_{q=1}^{Q} w_q^x(x_n) \left[\mu_q^y - \sum_q^{yx} \left(\sum_q^{xx} \right) - 1\left(x_n - \mu_q^x\right) \right]$$

$w_q^x(x_n)$: Probability source f rame belongs to acoustic class described by component q(caluculated in Decoding)

…(eq. 8)

3.2.1.B. Deep learning methods

In recent years we have seen the dominance of deep learning methods in solving previously complex computational problems in domains such as computer vision [31], speech recognition [12], etc. This success has motivated us to utilize the power of one of the simplest deep learning models, autoencoders[1], to learn voice conversion from a source to a given target speaker while translating the content information of the speech.

The simple Encoder-Decoder architecture of an autoencoder has been favoured by many researchers in the field of language translation and voice conversion, and has been the core technology in many state-of-the-art artificial intelligent engines such as the popular Google's translation engine since 2017 [36] and forms the bread-and-butter of most advanced sequence-to-sequence language models such as Tacotron, GTP and Tacotron 2.

3.2.2. Autoencoders

As stated previously, we use simple autoencoders for the task of voice conversion following the work of [3]. The DeepAI community defines an autoencoder as an unsupervised learning technique for neural networks that learns to efficiently represent data. This means that the network essentially learns to capture the signal and ignores the "noise" part of the training data.

Generally speaking, an autoencoder network has three main blocks or layers: the encoding block that receives input embeddings, a hidden layer, usually referred to as the bottleneck layer that extracts the principal components of the input, and finally the decoding block.
The ultimate objective of this network is to train itself on the input data and using backpropagation, find a low dimensional representation of the same input, i.e. the model learns important features of the input and reconstruct the same input in low dimension [2] while minimizing the reconstruction error equation (11) and equation (12).

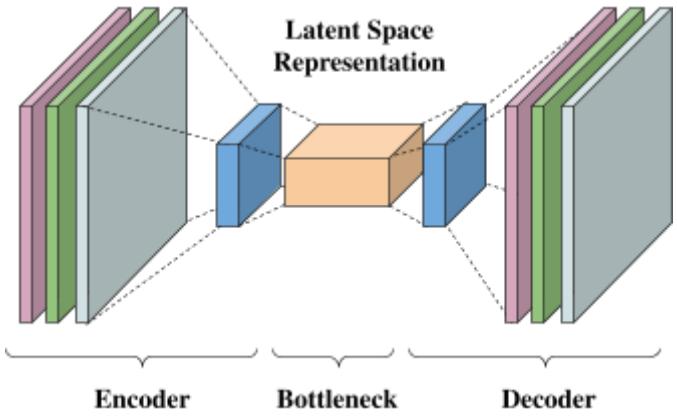

Figure 21: Autoencoder architecture

This process sometimes involves multiple autoencoders, such as stacked sparse autoencoder layers used in image processing as depicted in the figure above.

Applying this theory to our work, we want to utilise the simplicity of autoencoders to ultimately learn a function ***f*** that maps every word spoken by the source speaker, represented here by vector *s* into the embedding subspace of a selected target speaker *s target.* That is to say, using the function learned, we want to obtain similar sounding words form our target speaker that the source speaker initially uttered equation (10).

$$\forall w \in W,\ \forall s = S: \left| h_{speech}(f(s, w)) \right. = f\left(s_{target,}\ w\right)$$

…(eq. 9)

$$min_{B} \|X - XBB^{T}\|^{2} \qquad \grave{x}BB^{T} = \min_{s \in S} \|s - \grave{x}\|^{2}$$

$$\downarrow$$

$$Best\ rank - k\ approximation\ of\ X$$

…(eq. 10)

$$min_{E,D}\, Error(X, D(E(X))) \qquad D(E(\grave{x})) \approx \min_{s \in S} Error(s, \grave{x})$$

…(eq. 11)

To better understand how an autoencoder works, let's take a simple use-case in which they are mainly used. In image processing for instance, the first autoencoder process will learn to encode easy features of an image like the angles of a roof, while the second analyzes the first layer output to encode less obvious features like a door frame, the rest of the layers will further pick up other features necessary in describing what a house likes like.

3.2.2.A. Exemplar Autoencoder

We address the problem of voice conversion by learning an Autoencoder specific to a target speaker following the work of [3]. This approach proves reliable in capturing subtle properties of the target speaker such as the ambient context and environmental acoustics of the speaker; and stylistic prosody of the speech.

Unlike other methods that learn a generic base model for all target speakers and then fine-tune the learned model individually on a selected target speaker [27,28,29,30]; our method focus is on capturing important feature embedding of one particular speaker at a time, hence the term "Exemplar" [3].

Furthermore, the inability of these methods to expand and include not-seen data points at testing constitutes another obstacle for our problem setting.

3.2.3. Our Approach

Building on the limitations of statistical methods, the abundance of non-parallel data and the immense computing power we have available these days, we formulate this problem in a relatively simple sequence-to-sequence problem with three parts.

- **Part one:** Given a target speaker represented as an audio stream, we want to learn an autoencoder specific to that target speaker.
- **Part two:** we obtain the target's voice by projecting any input audio stream into the target's autoencoder as shown in Figure 23.
- **Part three:** we translate the source content from English to French.

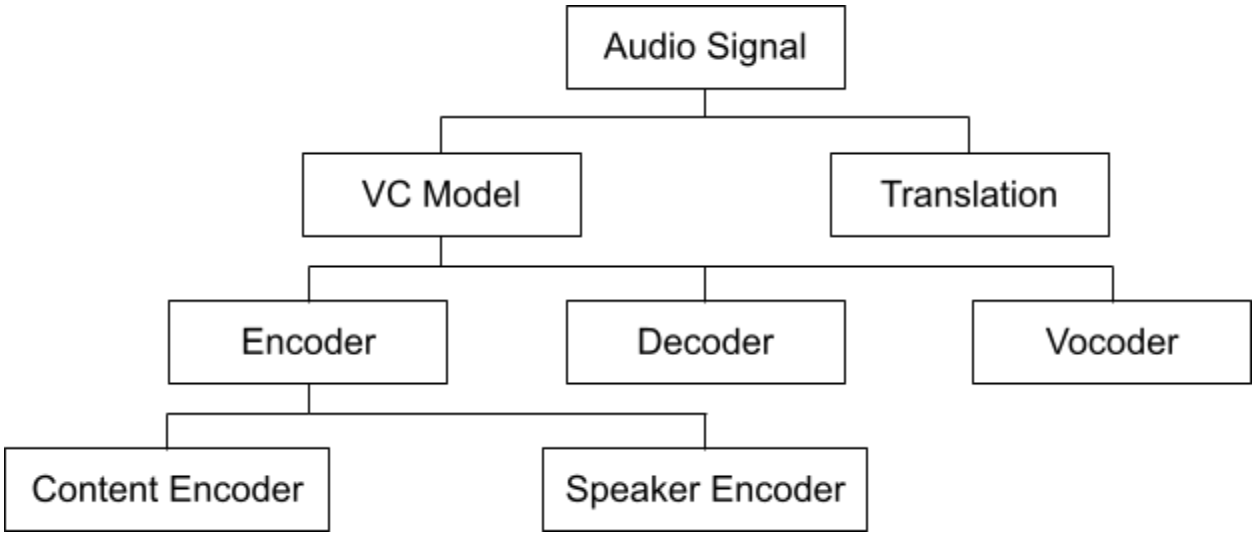

Figure 22: Voice and translation synthesis model overview.

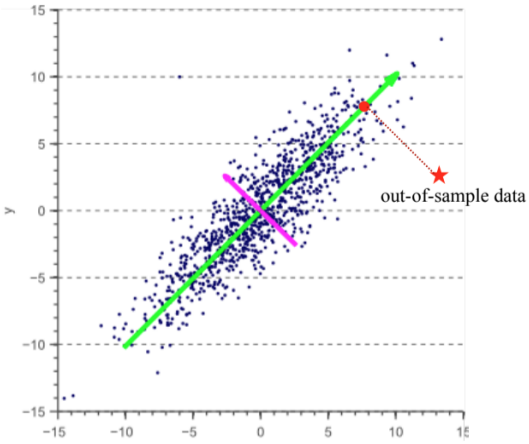

Figure 23: Linear or Exemplar autoencoder [3]

3.2.3.A. Encoder

For a generic text-based translation task the encoder has the role of mapping each word token in the input-sequence to a vector of fixed length commonly referred to as the context vector [3].

Our task is to use any audio speech as input to our model and not text, then find a way to translate and synthesize a natural speech as output.

Synthesizing natural speech requires training on a large number of high quality parallel audio samples, and in most cases, extending the model to support many speakers also would require even more parallel data and hundreds of hours of training per speaker [26].

Following the work of [16] in which the authors successfully demonstrate how transfer learning form speaker verification can be applied in a speech synthesis context while maintaining our encoder-decoder architecture [3], we proceed by decoupling or disentangling speaker and content modeling from the speech synthesis.

Doing this will help us train each component independently of each other and will reduce the need to obtain high quality multi-speaker training data. It consists of two sections as speaker encoder and content encoder.

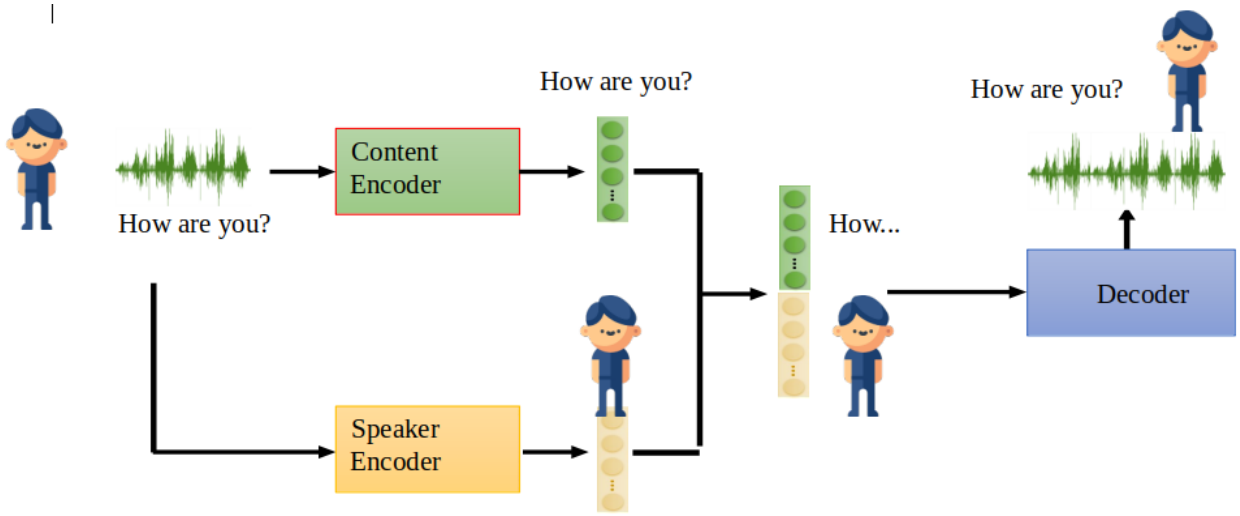

Figure 24: Network decoupling summary

Section 1: Speaker Encoder

The objective of the speaker encoder is to take source audio input sequence encoded as a Mel spectrogram frame, and output an embedding that contains everything characteristic of the speaker, i.e. how the speaker sounds (pitch, tone, accent, etc.) respective of its phonetic content.

Features learned from a particular speaker are combined into a low dimensional fixed-sized vector referred to as speaker embedding[3,27]. This network is trained to optimize a generalized end-to-end speaker verification loss so that embeddings of a speech from the same speaker appear relatively close to each other in terms of cosine similarity while those from a different speaker remain far apart in the embedding subspace.

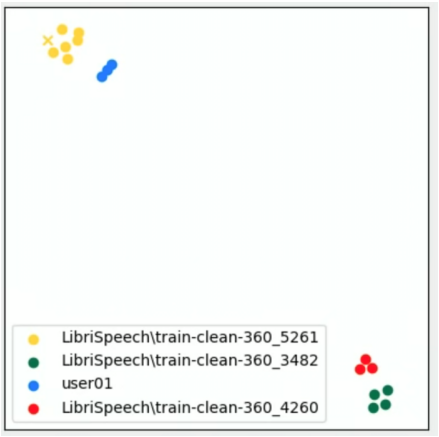

Figure 25: T-SNE for acoustic feature of words spoken by 4 speakers[2]

The speaker encoder model learns these embeddings in process described below:

1. The training audio samples are segmented into shorter 1.6 second clips of audio and transformed into log-mel spectrogram frames of 80 channels.

2. Input 80-channel log mel spectrograms are passed through a stack of 3 1D-CNNs with a kernel size of 5, each followed by batch normalization [36] and ReLU activation, and 2 Bi-LSTM layers of dimensions 32.
3. At training time, feature vectors of different speakers are assembled in a batch S, and where speakers has utterance U. Passing a feature vector xij, where I and I belong to the set S and U respectively; to the network we obtain a final embedding eij that is L2 normalized.
4. At inference time, the network takes two audio samples and decides whether or not they were spoken by the same speaker (using a softmax function). As a byproduct of this training, we find that this forces the speaker encoder to learn embeddings that represent how the speaker sounds by conditioning the synthesis network on the speaker identity and ignoring the content of the speech.

Section 2: Content Encoder

The content encoder network follows the same architecture of the speaker encoder except that, here the network is tuned to only capture the grapheme or phoneme sequence (the content) of the speech and ignore the speaker's characteristics following the works of Quan et al. [16] and [27]. Note here that the output of the content encoder is a sequence of word embeddings of fixed-length which will later be fed into the translation module ESPNet.

[2] https://medium.com/analytics-vidhya/the-intuition-behind-voice-cloning-with-5-seconds-of-audio-5989e9b2e042

3.2.3.B. Decoder

During training, the decoder receives as input the concatenation of both the content and speaker outputs and in turns tries to predict the target-sequence token by token as shown in Figure 24.

At inference time, by conditioning the output of the decoder to the output of the speaker encoder, we notice that, given a different speaker's content embedding the network is able to maintain the original speaker's voice with this speaker's content as depicted by Figure 26.

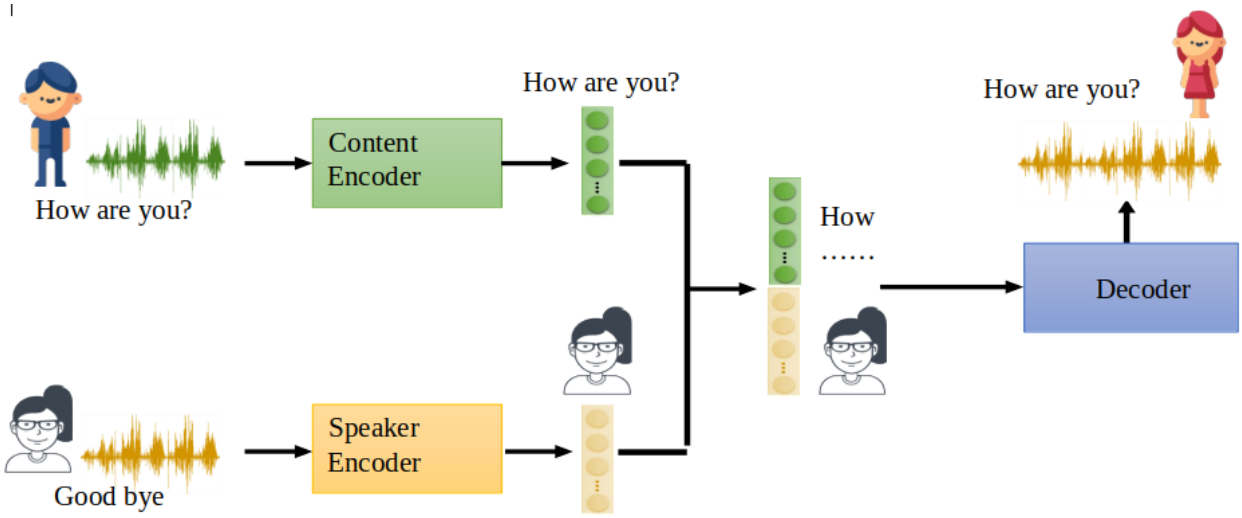

Figure 26: Network decoupling during inference

3.2.3.C. Vocoder

Because the decoder's outputs are Mel-spectrograms, we need to find a way to transform these spectral features into time-domain waveforms. We use the auto-regressive WaveNet [23] as a vocoder to invert Mel-spectrograms predicted by the decoder network into audio format.

As for the WaveNet architecture, we followed the one suggested by [23], this network has stacked residual blocks consisting of 30 dilated convolutional layers to directly model waveform samples however, we do not condition the output of the network of the speaker encoder but rather to the output of the decoder.

.

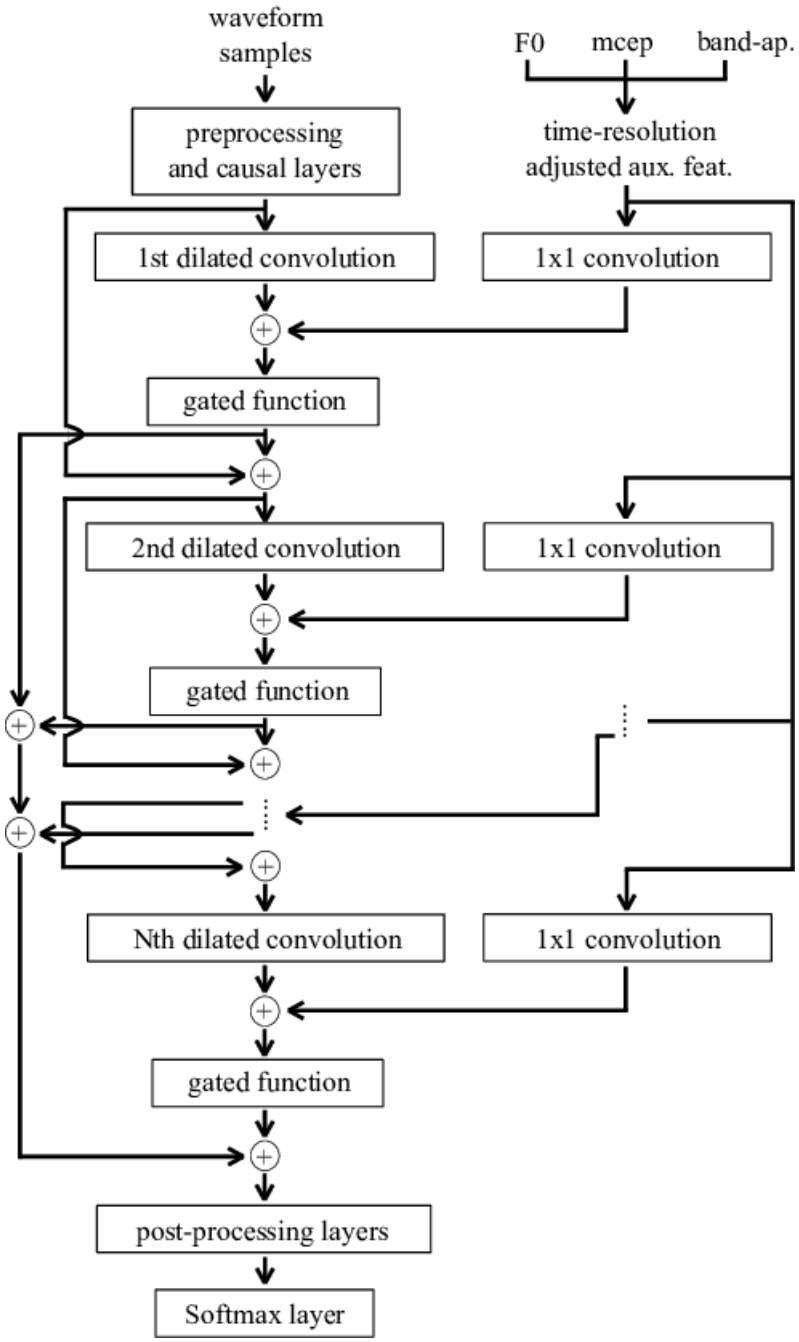

Figure 27: WaveNet model architecture to generate time-domain waveform given the estimated spectral parameters and linearly transformed F0 parameters[3].

[3] https://www.researchgate.net/publication/325997662_NU_Voice_Conversion_System_for_the_Voice_Conversion_Challenge_2018

First introduced by the DeepMind team in 2016 [34], WaveNet has become the defacto vocoder framework for most state-of-the-art conversion models. In principle, the architecture of WaveNet vocoder was motivated by the wide adoption and success of deep convolutional neural networks (CNNs) in computer vision and audio processing and therefore borrows its techniques heavily from the aforementioned domains. As seen in Figure 27, the network receives raw audio signal as input, passing it through multiple CNN layers before sampling a softmax over the distribution of the generated signal values. If trained from scratch, this process can take hours if not days to generate a 1 second of audio, reason why we utilized a pre-trained WaveNet Vocoder and adapted dimensions of its input layer to fit our requirements.

3.2.4. Model architecture

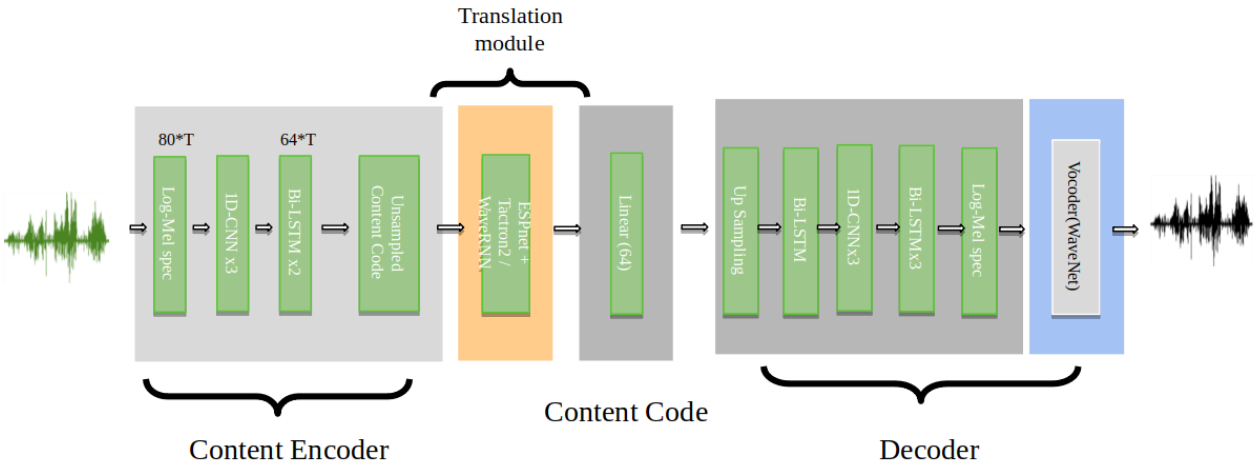

Figure 28: Network architecture

The network architecture of our model is built on the foundation laid down by most of the preceding works in voice conversion, and more specifically by three baseline papers we followed during the course of our work. In essence, we used the mainstream content encoder-decoder architecture for voice synthesis and extended it to include content translation.

Like [3] and [27], the content encoder network is a standard multi-layer CNN, but with added memory such that the layers carry over information with a simpler mapping function to the output channel.

We experiment with both 40 and 80 channels encoder architectures and obtain best results with an 80-channel speech Mel-spectrograms using short-time Fourier Transform. We modify and adapt layers to the spectrogram input and apply batch normalization before computing rectified linear unit (ReLU) activation.

Note here that, the use of the term content encoder encompasses or implies both the speaker and the content encoder networks because they essentially follow the same architecture except for some minor adaptations in the content encoder's bottleneck [3,16].

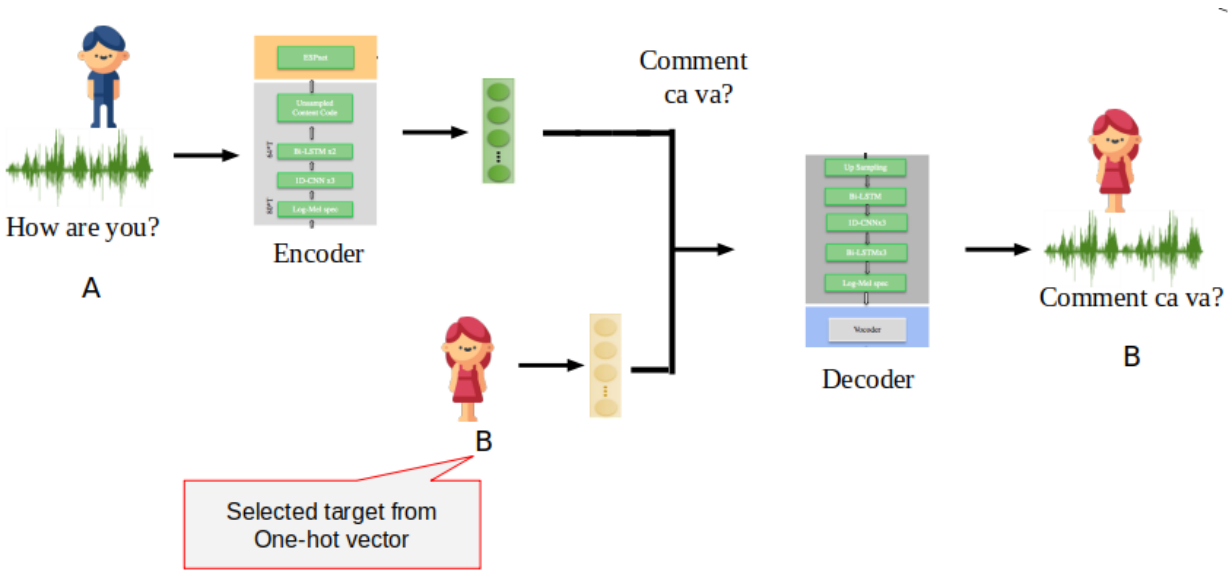

Figure 29: Summary of model architecture during inference time

The result of the content encoder is then passed through a pre-trained translation toolkit module ESPNet [1] as a raw and unsampled matrix containing the concatenated results of the two encoders as demonstrated in Figure 29 (as well as in Figure 26). Here also, the input layer of the ESPNet module has been modified to receive an unsampled content code as input instead of its usual raw audio signal.

The obtained embedding is flatten by a linear layer then up-sampled to the original time resolution and sequentially input to a 512-channel bi-directional LSTM layer and three layers of 512-channel 1D convolutional layers with a kernel size of 5 following [3]. Finally, the output is fed into three 1024-channel LSTM layers and a fully connected layer to project into 80 channels (Figure 28). The projection output is regarded as the generated Mel-spectrogram that is finally passed through the vocoder network and converted into time-domain waveform (Figure 27) as spoken by the our target speaker.

Chapter 4: Experiments

4.1. Data Pipeline and Foundation

For the translation part, we have a pretrained model on the Librispeech dataset which contains 236 hours of English read speech with its corresponding text in English and French translation. And moreover we added the English to spanish pretrained model based on the Fisher and Callhome spanish dataset as the multilingual translation choice. The Fisher and Callhome spanish dataset contains 170-hours of Spanish read speech with its corresponding Spanish text transcription and English text.

For Audiosynthesis, we have about 47 minutes audio of Barack Obama[4] , about 11 minutes audio of Bill Clinton's speech and about 50 minutes audio of Oprah Winfrey[5]. We trained the targeted speaker model with 2000 iterations of batch-size 8 and epochs 10. As the machine setting, we used the Google Colab pro for the experiment. So we could access either T4 or P100GPU with high memory.

4.2. Metrics

We used BLEU (bilingual evaluation understudy) scores to see the accuracy of the translation by comparing between the ESPnet generated french transcript and french grandtruth (formula 13). Machine translation output quality is evaluated by BLEU score. Bleu score is graded between 0 to 100 and 100 is the highest value of the accuracy and 0 is the viceversa.

$$BLEU = BP_{BLEU} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$P_n = \frac{\sum_i \text{Number of } n-\text{ grams matched between translated sentence } i \text{ and ref erence sentence } i}{\sum_i \text{ Number of all } n-\text{ grams in the translated sentence } i}$$

$$W_n = \frac{1}{N}$$

…. (eq. 12)

BP BLEU is a penalty when the translated text is shorter than the reference translation. If the translated text is longer, BP BLEU is 1 and there is no penalty. N-gram means "N consecutive characters next to each other", e.g. 2-gram for 2 characters.

[4] https://www.youtube.com/watch?v=ji6pl5Vwrvk

[5] https://www.youtube.com/watch?v=GR_7X0exvh8

As the comparison of the translation accuracy between our model and other researches, we use Unsupervised attentional encoder-decoder + BPE [37](2018) (an unsupervised attentional encoder and a decoder model) and Unsupervised NMT + Transformer [38](2018) (transformer) as the benchmarks.

Unsupervised attentional encoder-decoder + BPE [37] is a standard encoder-decoder architecture with an attention mechanism. They use a two-layer bi-directional RNN in the encoder, and another two-layer RNN in the decoder. It handles dual translation direction (e.g. English → French, French → English). Unsupervised NMT + Transformer [38] is NMT models built upon Standard LSTM and Transformer. It handles dual translation directions (e.g. English → French, French → English).

To evaluate audio synthesis, we conduct listening tests to measure subjective speech quality. We conducted the mean opinion score to 30 people on our social media. And the mean opinion score grades are 1 to 5. As the evaluation criteria, we adopt naturalness of the audio, voice similarity of the audio to the targeted speaker and content consistency how well the synthesized audio could keep the content of what the target speaker is saying (Table 2). And as the benchmark, we use GAN with wavenet [39](2018). GAN with wavenet [39] uses a CycleGAN to produce a characteristic similar voice to the target speaker.

Naturalness	Naturalness of the audio
Voice Similarity	Voice similarity of the audio to the targeted speaker (Ground Truth).
Content Consistency	How well the synthesized audio could keep the content of what the target speaker is saying.

Table 2: Evaluation of the audio

The Mean Opinion Score (MOS) questions for evaluating the Naturalness, Voice Similarity and Content Consistency are listed below. Survey conducted to compare our model and related papers performance of Audio-synthesis.

- Naturalness: (On a generated audio)
  - How natural is this recording?
- Voice Similarity: (Ground truth vs. generated audio)
  - Are these 2 audios from the same speaker?
- Content Consistency: (Ground truth vs. generated audio)
  - Are these 2 people saying the same thing?

4.3. Experimental Result

Table 3 shows the BLEU score comparison between our model and the benchmark model. Our model has a higher score than the unsupervised attentional encoder-decoder model but lower than the recent research with transformers.

	BLEU Score
Unsupervised attentional encoder-decoder + BPE [37]	14.36
Unsupervised NMT + Transformer [38]	25.14
Our Model	**18.20**

Table 3: Comparison of BLEU Scores

The graph in Figure 30 shows how well the autoencoder can reconstruct and get close to the ground truth audio by the voice conversion loss[6]. As we mentioned earlier, we set the batch-size 8 and epochs 10 to train the model. At the end of the training, we used Tensorboard to obtain the graph. As you can see the graph, the convergence of the total loss rapidly decreased and after 200 iterations the convergence became slow. As a result, the total loss settled down at around 0.2.

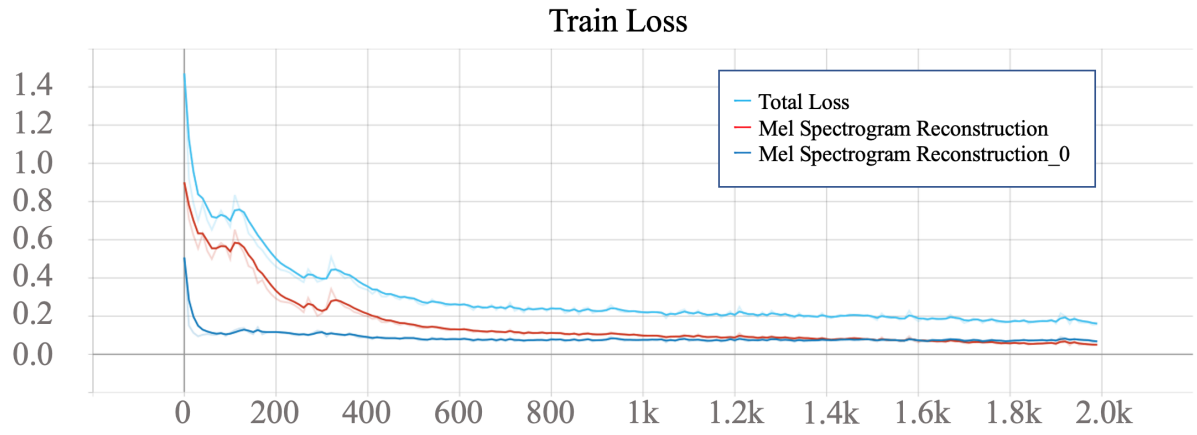

Figure 30: The voice conversion loss of our Obama model

[6] https://github.com/Student-Research-Project/Speech-Synthesis-and-Translation-via-Exemplar-Autoencoders

The below formula shows the calculation error between the imputed features of x and the generated speech of tilde x and the ground truth of m and reconstruction of tilde m.

$$\text{Error}_{audio} = \mathbb{E}\|x - \tilde{x}\|_1 + \mathbb{E}\|m - \tilde{m}\|_1$$

…(eq. 13)

Tables 4 and Table 5 show the MOS of our model compared to the other speech synthesis models such as the GAN with wavenet and the Unsupervised any to many model. For the accurate comparison, we generated English to English audios using the ground truth audios from the demo pages of GAN with Wavenet [39] and Unsupervised Any-to-Many[1] respectively. And we compared our generated audios to the same context of English to English generated audios of GAN with wavenet [39] and Unsupervised Any-to-Many [1].

The parentheses 1 in the Table 4 and 5 show the result of English to English generated audio by our model. And as the comparison results of English to English generated audios (Table 4), we could ensure that our synthesis model works better than the GAN with the Wavenet model.

We could also confirm that the performance of our synthesis model is as high quality as the unsupervised any to many model [1] by the parentheses 1 in the Table 5. With this synthesis model, we generated English to French translated and synthesized audio. The MOS of generated English to French audio is shown in the parentheses 2 on each Table 4 and Table 5. As expected, our English to French audio synthesis MOS is lower than the MOS of our English to English generated audio, especially the content consistency.

However we could confirm that our generated English to French audio naturalness and voice similarity is more natural than the generated English to English audio of the GAN with wavenet and sustains the similarity to the targeted speaker's ground truth more than the English to English generated audio of the GAN with wavenet. (Mention different data used in Table 4 and Table 5 for comparing to our model because our model numbers are different in the tables)

Ground Truth of Obama:	**Naturalness ↑**	**Voice Similarity ↑**	**Content Consistency ↑**
GAN with Wavenet[39]:	2.35 ± 0.95	2.12 ± 1.14	3.15± 0.89
Our Model:	**(1)3.21 ± 0.89** **(2)2.36 ± 1.01**	**(1)3.29 ± 1.02** **(2)2.64 ± 0.74**	**(1)3.21±0.91** **(2)2.9 ±1.04**

Table 4: Comparison of Mean Opinion Score (Obama)

Ground Truth of Oprah:	**Naturalness ↑**	**Voice Similarity ↑**	**Content Consistency ↑**
Unsupervised Any-to-Many [3]:	2.78 ± 0.70	3.68 ± 0.98	3.32 ± 0.82
Our Model:	**(1) 2.45 ± 1.22** **(2) 1.96 ± 0.48**	**(1) 3.34 ± 1.10** **(2) 2.41 ± 0.52**	**(1)3.13±0.99** **(2)1.42±0.61**

Table 5: Comparison of Mean Opinion Score (Oprah)

Chapter 5: Discussion

Our model consists mainly of two tasks: Translation and Audio synthesis.

First, as a baseline paper for our translation model, we chose the two papers, ESPnet-ST: All-in-One Speech Translation Toolkit [1], One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech [2]. With the first paper techniques, we could enable our model to translate from English speech to French text and with the second, we could incorporate the audio synthesis techniques of French text to French speech to our model. That means our translation is a cascade model.

As we introduced the paper called Direct speech-to-speech translation with a sequence-to-sequence model [4], there are direct (end to end) models of speech to speech translation. Naturally it would be better to select this paper as our baseline. However we had a resource problem, that is, to implement the algorithm of speech to speech translation, we had to prepare a dataset of English and French pair speeches of the same targeted speaker. In that respect of the difficulty to collect the dataset, we choose the cascaded speech to speech translation model to implement. If we had the data resources, we would choose the direct model of speech to speech translation.

About the dataset used in the translation, the accuracy of our translation model depends on the librispeech dataset created by the speech reading the novel and its French translation. Since Neural Machine Translation is dependent on the training data, we think that it would have been better to use other training data instead of just Libri speech.

As for the evaluation of translation accuracy, it is necessary to consider more evaluation methods than just the BLEU score, since the setting of n-grams is not meaningful to a randomly selected sentence. That is because the BLEU score is a corpus-based indicator and we used it as a metric to evaluate individual sentences and collected the statistics across the individual sentences. Also, the BLEU score indicator doesn't evaluate the grammar properly even if the meaning of the sentence is correct. The penalty is the same even if one letter is left out, resulting in a completely different sentence.

Secondly, as the baseline of our audio synthesis model, we chose a paper named Unsupervised Any-to-Many Audiovisual Synthesis via Exemplar Autoencoders [3]. With the techniques from the papers, we could enable our model to synthesize audio while maintaining the features of the target speaker. And in fact, our model was able to achieve a higher average opinion score when comparing the English speech produced by our model to the speech produced by the baseline paper model while sustaining the naturalness, voice similarity and content consistency of the targeted speaker.

However, when using the same model with French speech as the impute, it was confirmed that the model was comparably able to maintain the similarity of the target speaker's speech, but was comparably unable to maintain naturalness and content consistency.

The problem, we believe, is that the autoencoder model trained on English data loses the naturalness and content consistency of French due to the difference in word sound between English and French when projecting French speech input. This problem makes it very difficult to adjust the parameters of French naturalness, content consistency and voice similarity, because if we seek French naturalness and content consistency, we may lose the voice similarity of the target speaker. It is also possible that this problem can be solved by using general bilingual speakers instead of celebrities as target speakers for the training data. However, we sought to solve the problem by using the voices of celebrities. The reason is that if the target speaker is a random French speaker, it is difficult to validate the generated speech since it cannot be compared with the baseline in the mean opinion score.

An interesting future orientation of our work could be altering the module structure used in our approach. This work could further be extended by eliminating TTS synthesis in the translation part. In order to do this, French speech output of a target speaker should be obtained directly after the ST module. By applying the idea to our model, we can reconstruct the translation model and audio synthesis model by content encoder to translate English speech to French and speech encoder to project the translated french speech to the targeted embedding space. With this process, we can get the vectorized features of a target speaker. And to get the speech signal, we construct a decoder to generate Mel spectrograms. Then the following vocoder can generate speech signals from Mel spectrograms. This approach might make our model more compact and efficient in terms of speed.

Chapter 6: Conclusion

In conclusion, we have implemented a cascaded speech translation which combines ESPnet, Tactron2 and WaveRNN. Our architecture consists of the implementation of speech to speech translation and the implementation of speech synthesis with an autoencoder. This allows the target person to speak a given language without knowing it. With our model, we could generate a better English to English synthetic voice than GAN with Wavenet [39]. However, we couldn't generate a better naturalness and content consistency of English to French synthetic voice than the generated English to English audio of naturalness and content consistency by the GAN with Wavenet [39] while sustaining the balance of Naturalness, Voice Similarity and Content consistency. On the other hand, we could generate a better voice similarity of English to French synthetic voice than the generated English to English audio. The architecture is also adaptable to different languages, which we believe is a game changer in the field such as the media industry or entertainment industry.

References

[1] Inaguma, H., Kiyono, S., Duh, K., Karita, S., Soplin, N. E. Y., Hayashi, T., & Watanabe, S. (2020). ESPnet-ST: All-in-one speech translation toolkit. *arXiv preprint arXiv:2004.10234*.

[2] Nekvinda, T., & Dušek, O. (2020). One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech. *arXiv preprint arXiv:2008.00768*.

[3] Deng, K., Bansal, A., & Ramanan, D. (2020). Unsupervised Any-to-Many Audiovisual Synthesis via Exemplar Autoencoders. *arXiv preprint arXiv:2001.04463*.

[4] Jia, Y., Weiss, R. J., Biadsy, F., Macherey, W., Johnson, M., Chen, Z., & Wu, Y. (2019). Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv preprint arXiv:1904.06037*.

[5] Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., ... & Rondon, P. (2019). Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*.

[6] Kuchaiev, O., Ginsburg, B., Gitman, I., Lavrukhin, V., Case, C., & Micikevicius, P. (2018, July). Openseq2seq: extensible toolkit for distributed and mixed precision training of sequence-to-sequence models. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)* (pp. 41-46).

[7] Zeyer, A., Alkhouli, T., & Ney, H. (2018). RETURNN as a generic flexible neural toolkit with application to translation and speech recognition. *arXiv preprint arXiv:1805.05225*.

[8] Zenkel, T., Sperber, M., Niehues, J., Müller, M., Pham, N. Q., Stüker, S., & Waibel, A. (2018). Open Source Toolkit for Speech to Text Translation. *Prague Bull. Math. Linguistics*, *111*, 125-135.

[9] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., ... & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

[10] Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., ... & Uszkoreit, J. (2018). Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

[11] Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

[12] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (No. CONF). IEEE Signal Processing Society.

[13] Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., ... & Collobert, R. (2019, May). Wav2letter++: A fast open-source speech recognition system. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6460-6464). IEEE.

[14] Tu, T., Chen, Y. J., Yeh, C. C., & Lee, H. Y. (2019). End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning. *arXiv preprint arXiv:1904.06508*.

[15] Lee, Y., Shon, S., & Kim, T. (2018). Learning pronunciation from a foreign language in speech synthesis networks. *arXiv preprint arXiv:1811.09364*.

[16] Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., ... & Wu, Y. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558*.

[17] Prakash, A., Thomas, A. L., Umesh, S., & Murthy, H. A. (2019, September). Building

Multilingual End-to-End Speech Synthesisers for Indian Languages. In *Proc. of 10th ISCA Speech Synthesis Workshop (SSW'10)* (pp. 194-199).
[18] Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Chen, Z., Skerry-Ryan, R. J., ... & Ramabhadran, B. (2019). Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. *arXiv preprint arXiv:1907.04448*.
[19] Taigman, Y., Wolf, L., Polyak, A., & Nachmani, E. (2017). Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*.
[20] Liu, Z., & Mak, B. (2019). Cross-lingual multi-speaker text-to-speech synthesis for voice cloning without using parallel corpus for unseen speakers. *arXiv preprint arXiv:1911.11601*.
[21] Cao, Y., Wu, X., Liu, S., Yu, J., Li, X., Wu, Z., ... & Meng, H. (2019, May). End-to-end code-switched tts with mix of monolingual recordings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6935-6939). IEEE.
[22] Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., & Bengio, Y. (2017). Char2wav: End-to-end speech synthesis.
[23] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Wu, Y. (2018, April). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4779-4783). IEEE.
[24] Abe, M., Nakamura, S., Shikano, K., & Kuwabara, H. (1990). Voice conversion through vector quantization. *Journal of the Acoustical Society of Japan (E)*, *11*(2), 71-76.
[25] Stylianou, Y., Cappé, O., & Moulines, E. (1998). Continuous probabilistic transform for voice conversion. *IEEE Transactions on speech and audio processing*, *6*(2), 131-142.
[26] Desai, S., Raghavendra, E. V., Yegnanarayana, B., Black, A. W., & Prahallad, K. (2009, April). Voice conversion using artificial neural networks. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3893-3896). IEEE.
[27] Qian, K., Zhang, Y., Chang, S., Yang, X., & Hasegawa-Johnson, M. (2019, May). Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning* (pp. 5210-5219). PMLR.
[28] Kaneko, T., Kameoka, H., Tanaka, K., & Hojo, N. (2019). StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion. *arXiv preprint arXiv:1907.12279*.
[29] Kaneko, T., Kameoka, H., Tanaka, K., & Hojo, N. (2019, May). Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6820-6824). IEEE.
[30] Fang, F., Yamagishi, J., Echizen, I., & Lorenzo-Trueba, J. (2018, April). High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5279-5283). IEEE.
[31] Forsyth, D. A., & Ponce, J. (2012). *Computer vision: a modern approach*. Pearson,.
[32] Kameoka, H., Kaneko, T., Tanaka, K., & Hojo, N. (2019). ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(9), 1432-1443.
[33] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., ... & Ochiai, T. (2018). Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*.
[34] Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... &

Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.
[35] Schuster, M., Johnson, M., & Thorat, N. (2016). Zero-shot translation with Google's multilingual neural machine translation system. *Google AI Blog*, *22*.
[36] Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., & Bengio, Y. (2016, March). Batch normalized recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2657-2661). IEEE.
[37] Artetxe, M., Labaka, G., Agirre, E., & Cho, K. (2017). Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.
[38] Lample, G., Ott, M., Conneau, A., Denoyer, L., & Ranzato, M. A. (2018). Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.
[39] Lorenzo-Trueba, J., Fang, F., Wang, X., Echizen, I., Yamagishi, J., & Kinnunen, T. (2018). Can we steal your vocal identity from the Internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data. *arXiv preprint arXiv:1803.00860*.