UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

End-to-end Speech Translation with Self-supervised Speech Representations

A Master's Thesis submitted to the Faculty of the

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya

by

Gerard Ion Gállego Olsina

In partial fulfilment of the requirements for the

Master's degree in Advanced Telecommunication Technologies

Supervisor:
Marta R. Costa-jussà

Barcelona, September 2020

"You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future."

Steve Jobs

Abstract

Nowadays, there is a growing interest in the field of Speech Translation (speech-to-text). Traditionally, this task has been faced with the concatenation of Automatic Speech Recognition and Machine Translation modules. Nevertheless, in the last few years, many researchers have proposed the use of an end-to-end approach, in which the speech is not transcripted but directly translated into the target language.

Furthermore, there is a notable research trend in the use of self-supervision techniques to train speech encoders. These systems do not need human-annotated data for training, and they can extract much richer speech representations than other traditional methods.

In this project, we explored the use of three pre-trained speech encoders (PASE+, APC and Wav2Vec) to improve end-to-end ST, using a Transformer as the core of our model. We trained it with the English-French split of the MuST-C corpus, comprising 492h of speech, and we developed our code on top of Fairseq, creating a repository which will facilitate our group's future research in ST.

Our system did not achieve the results of the baseline, but we think that there is still room for improvement, and we believe we will be able to compete with state-of-the-art ST systems using pre-trained speech encoders in the future.

Resumen

Actualmente hay un creciente interés en el campo de la traducción del habla (de voz a texto). Tradicionalmente, esta tarea se ha afrontado concatenando módulos de reconocimiento de voz y de traducción automática. Sin embargo, en los ultimos años muchos investigadores han propuesto el uso de una estructura de traducción directa, en la cual la voz se traduce sin necesidad de obtener la transcripción intermedia.

Además, hay una importante corriente de investigación en el uso de técnicas de autosupervisión para entrenar codificadores del habla. Estos sistemas no necesitan datos etiquetados por humanos para su entrenamiento, y pueden extraer representaciones del habla mucho más ricas que otros métodos tradicionales.

En este proyecto, hemos explorado el uso de tres codificadores pre-entrenados (PASE+, APC y Wav2Vec) para mejorar los sistemas de traducción directa del habla, usando un Transformer como componente central de nuestro modelo. Lo hemos entrenado con la partición Inglés-Francés del corpus MuST-C, que contiene 492h de voz, y hemos desarrollado nuestro código sobre Fairseq, creando un repositorio que facilitará la investigación futura de nuestro grupo en la traducción del habla.

Nuestro sistema no ha conseguido equipararse con los resultados del sistema de referencia, pero pensamos que aún tenemos margen de mejora, y creemos que en el futuro seremos capaces de competir con sistemas punteros, usando codificadores del habla pre-entrenados.

Resum

Actualment, hi ha un interès creixent en el camp de la traducció de la parla (de veu a text). Tradicionalment, aquesta tasca s'ha resolt concatenant mòduls de reconeixement de veu i de traducció automàtica. Ara bé, els últims anys molts investigadors han proposat l'ús d'una estructura de traducció directa, en la que la veu es tradueix sense necessitat d'obtenir la transcripció intermitja.

A més, hi ha un important corrent d'investigació en l'ús de tècniques d'autosupervisió per entrenar codificadors de la parla. Aquests sistemes no necessiten dades etiquetades per humans per entrenar-se, i poden extreure representacions de la parla molt més riques que altres mètodes tradicionals.

En aquest projecte, hem explorat l'ús de tres codificadors pre-entrenats (PASE+, APC i Wav2Vec) per millorar els sistemes de traducció directa de la parla, utilitzant un Transformer com a component central del nostre model. L'hem entrenat amb la partició Anglès-Francès del corpus MuST-C, que conté 492h de veu enregistrada, i hem desenvolupat el nostre codi sobre Fairseq, creant un repositori que facilitarà la investigació futura del nostre grup en aquest camp.

El nostre sistema no ha aconseguit equiparar-se amb els resultats del sistema de referència, però pensem que encara tenim marge de millora, i creiem que en el futur serem capaços de competir amb sistemes punters, utilitzant codificadors de la parla pre-entrenats.

A la Laura,
per fer-me riure els dies més difícils.

Acknowledgements

I would like to thank Marta R. Costa-jussà, for introducing me to the fascinating world of research, and for the great effort she is making to continue growing as a group. To Carlos Escolano, for being such a great teammate, and for the time he spent helping me. And to Guillem Cortès, who was an excellent classmate during the master's degree and has become a real friend during the last months.

I would also like to thank my parents for showing me the value of happiness and for their support all these years.

And finally, I would like to especially thank Laura's parents, who hosted me in their home during the Covid-19 lockdown. I could not be more grateful for it.

Revision history and approval record

Revision	Date	Purpose
0	15/06/2020	Document creation
1	01/09/2020	Document approval

Name	e-mail
Gerard I. Gállego	gerard.ion.gallego@upc.edu
Marta R. Costa-jussà	marta.ruiz@upc.edu

	Written by	Reviewed and approved by
Date	01/09/2020	01/09/2020
Name	Gerard I. Gállego	Marta R. Costa-jussà
Position	Author	Supervisor

Table of contents

List of Figures

List of Tables

List of Abbreviations

ASR	Automatic Speech Recognition
BLEU	BiLingual Evaluation Understudy score
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficients
MT	Machine Translation
QRNN	Quasi-Recurrent Neural Network
RNN	Recurrent Neural Network
ST	Speech Translation
TTS	Text-To-Speech
WER	Word Error Rate

Chapter 1

Introduction

We, the humans, are heterogeneous. We organise ourselves in diverse societies, with many cultures, distinct gods, and multiple languages. This diversity among human beings is what makes each of us unique. We, the researchers, have the responsibility to build strong bridges to ensure dialogue between humans, since we can help breaking language barriers and improve people's communication. Still, we are in a double-edged sword situation, since we may be destroying, in the long run, all those languages which are not available yet in our translation systems, and with them, a part of the human diversity.

Motivation This master's thesis was developed as part of a project that is being carried out at the UPC Machine Translation group. The goal of AMALEU[1] is to build a highly multilingual translation system, with an architecture designed in a way to ease the extension to new languages, and in which low-resourced languages can benefit from the others. Currently, the system is working with text-to-text translation, but we would like to extend it to be able to translate speech too.

The initial purpose of Speech Translation[2] systems was to translate non written languages [Besacier et al., 2006], but nowadays they are also useful for real-time communication or live streaming subtitling. Traditionally, this task has been faced with the concatenation of Automatic Speech Recognition and Machine Translation modules, conforming what is known as a cascade system. Nevertheless, in the last few years, many researchers have proposed the use of an end-to-end approach in which the speech is not transcripted but directly translated into the target language (Fig. 1.0.1).

The end-to-end method, which is based in the encoder-decoder architecture [Cho et al., 2014][Sutskever et al., 2014], fits perfectly into the system developed in AMALEU. Moreover, as many researchers have pointed out the last few years, it supposes many advantages over the cascade approach, such as lower inference time, the direct use of prosodic from speech, or

[1] More information at: https://futur.upc.edu/25870473

[2] Speech Translation refers to speech-to-text translation.

the avoidance of the error concatenation, a classical problem of cascade systems. However, end-to-end systems still struggle to achieve pipeline performance due to data scarcity, and it exists a tradeoff between the error propagation of the cascade approach and the data inefficiency of the end-to-end method, as pointed out by [Sperber et al., 2019].

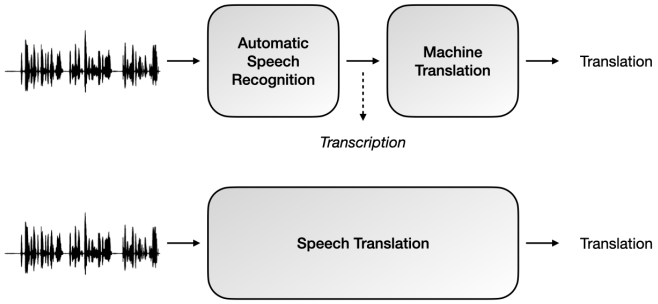


Figure 1.0.1: Speech Translation approaches.
Cascade (top) & End-to-end (bottom).

Objectives This project has three main goals. The first one is to land on the Speech Translation field, doing an in-depth literature review to set the basis of our future research in this field. The second goal is to produce a reliable and modular code to reduce the amount of work needed for launching new experiments in the future. And finally, the third objective is to explore the use of self-supervised speech representations to improve end-to-end ST results and help to close the gap between this method and the cascade approach.

Contributions We investigated the use of three pre-trained self-supervised speech encoders: PASE+ [Ravanelli et al., 2020], APC [Chung and Glass, 2020] and Wav2Vec [Schneider et al., 2019]. Some researchers have already demonstrated their efficacy in ST [Wu et al., 2020], but no one has explored the use of them with a Transformer model yet [Vaswani et al., 2017]. The code we developed is built on top of Fairseq [Ott et al., 2019], a sequence-to-sequence toolkit from Facebook AI Research, based on PyTorch [Paszke et al., 2019]. For some of our experiments, we also used pre-trained components from NeMo [Kuchaiev et al., 2019] and from OpusMT [Tiedemann and Thottingal, 2020].

Thesis organisation This dissertation begins with a review of the theoretical background of the sequence-to-sequence architectures, which are the base of our system; and self-supervision methods to obtain speech representations (§2). After that, we describe the state of the art of end-to-end ST, with a literature review of the systems that have been proposed and the most important corpora that exist nowadays (§3). In the Methodology chapter, we describe the system we developed, as well as the ones we used as a baseline (§4). Then, we explain our experimental framework, with a description of the data preprocessing, our code implementation and the parameters we used (§5). We present and discuss the results in §6, and we close the dissertation with our conclusions and an exposition of the future experiments that we will carry out during my PhD (§7).

Chapter 2

Background

In this project, we used advanced deep learning techniques; hence, in this chapter, we introduce key concepts which are the base of the methods we used for our experiments.

First, we describe important concepts about sequence-to-sequence architectures, in which our system is based. We review the motivations that lead researchers to use this scheme for end-to-end Speech Translation, and we describe the evolution of this architecture, including a description of the Transformer model, the core of our system.

Second, we introduce the pre-trained speech encoders that we used as feature extractors. We start reviewing the classical approaches for extracting features from the speech; and then, we introduce the concept of self-supervision and its use to extract speech representations. We close this chapter with an explanation of the three speech encoders we used in our work.

2.1 Sequence-to-sequence architectures

The most naive approach when facing an end-to-end ST task is to train an Automatic Speech Recognition model with translations instead of transcriptions. Nevertheless, there is a crucial concept which makes ST nearer to Machine Translation than ASR, despite having speech inputs. While ASR consists of detecting which character the speaker is uttering at each instant, MT systems need to align source and target words besides translating them. For this reason, a common approach to build end-to-end ST systems is to adapt an MT model to accept speech at the input [Di Gangi et al., 2019b].

The encoder-decoder systems proposed by [Cho et al., 2014] and [Sutskever et al., 2014] for Machine Translation became a cornerstone for this field. These architectures were based in recurrent neural networks, a type of network whose outputs depend on the input and on previous states of itself. This temporal sense makes them perfect for processing sequences of data, like text or audio. The encoder is in charge of processing a sample at each time step (e.g. words of a sentence), gathering information of the whole input sequence and collapsing

it in a final representation, the context vector. Then, the decoder uses this vector to condition an autoregressive generation of the output sequence (Fig. 2.1.1).

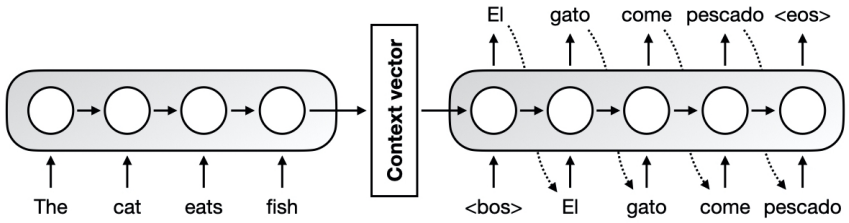


Figure 2.1.1: Sequence-to-sequence architecture for MT.

However, some researchers realised that a single vector could not retain enough information to represent a whole sentence. Hence, [Bahdanau et al., 2015] proposed the *attention mechanism*, that consists in adding an auxiliary network which learns how to combine all the encoder intermediate states in such a way that the decoder can take into account multiple input words. This method supposed a huge step forward in Machine Translation [Luong et al., 2015] and has deeply influenced the research in the field.

The generalised use of RNNs in MT gave way to some approaches with convolutional neural networks [Gehring et al., 2017], whose usage was more extended in the Computer Vision field. Nevertheless, the most significant advance in MT came with the creation of the *Transformer* [Vaswani et al., 2017], which brought a massive revolution in NLP [Devlin et al., 2018][Radford et al., 2019][Brown et al., 2020]. The Transformer is based in the encoder-decoder architecture, which allowed researchers to use it effectively to other domains beyond the text processing, like speech [Dong et al., 2018] or image [Parmar et al., 2018].

The authors of this model proposed an innovative idea, instead of relying on recurrence, they introduced the use of *self-attention*. This method consists in applying the attention mechanism to the sequence itself, extracting information about the relationships between its elements. Moreover, they went further and used *multi-head self-attention*, that consists of using multiple self-attention networks in parallel and combine their output.

While the self-attention architecture allowed much faster training by getting rid of recurrence, it does not retain information of the sequence order. For this reason, it is necessary to sum *positional embeddings* at the input of the Transformer encoder and decoder, normally based in sine and cosine functions.

Similar elements compose the encoder and the decoder. Both of them consist of a stack of layers containing a multi-head self-attention module, residual connections, normalization and a feed-forward network. Besides, the decoder adds some variations to the encoder architecture. First, self-attention is masked to avoid attending future samples, since it must be autoregressive. Second, the decoder layers also contain attention modules whose objective is to attend to encoder outputs, that is, the classic attention method.

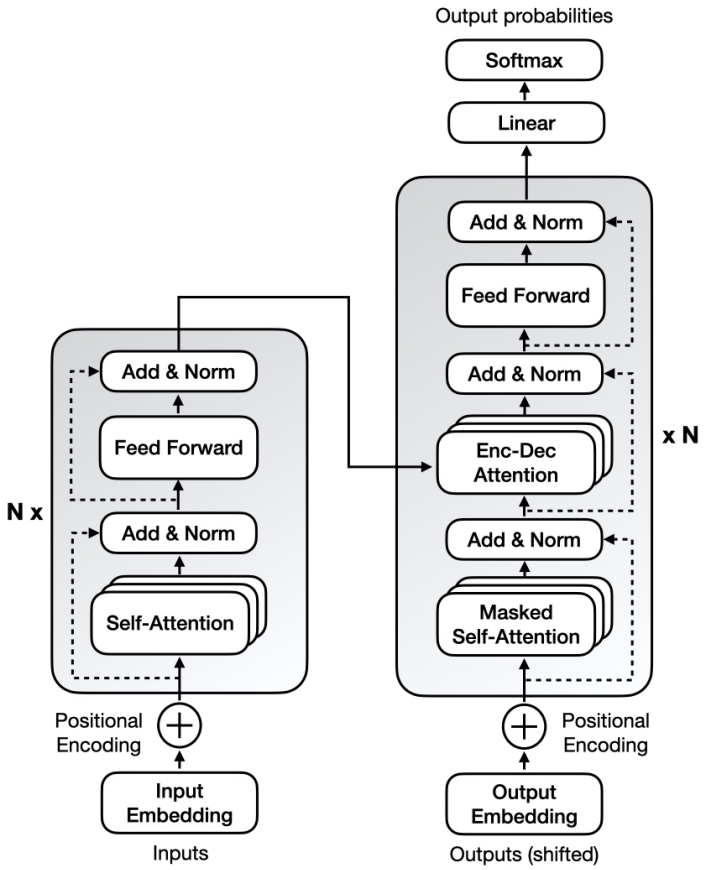


Figure 2.1.2: Architecture of the Transformer.

2.2 Self-supervised speech representations

Using the raw audio waveform directly is not a common approach when working in speech-related tasks, and the most frequent is to extract features from the audio beforehand. These representations usually are spectrogram-based, so that the 1-dimensional waveform is transformed into two dimensions representing time and frequency information (Fig. 2.2.1).

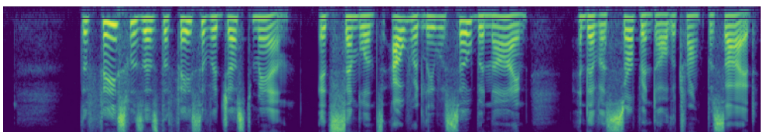


Figure 2.2.1: Log-Mel spectrogram of a speech fragment.

These representations have some advantages over working with the speech waveform. The most important one is that, since they are computed over audio frames, they drastically reduce the sequence length. They also contain meaningful information to understand human voice and their computation is low.

However, recently, some researchers claimed that these features are not representative enough of all the acoustic and linguistic properties of the human voice. Hence, they proposed the

use of self-supervised deep learning methods to obtain richer representations that contain high-level properties of the voice.

Self-supervision is a subtype of unsupervised learning (i.e. machine learning tasks without human-labelled data) where targets are computed from the input data itself [Doersch et al., 2015][Misra et al., 2016]. Taking into account that human-labelled data is much more difficult to collect than unlabelled data, self-supervised methods allow to pre-train components with much more data, and improve results of supervised downstream tasks [Devlin et al., 2018].

In this project, we examined three different self-supervised speech encoders: PASE+, APC and Wav2Vec. From now on, we review each of these approaches, briefly describing its architecture, the labels extracted from the input data, and the datasets used for training them.

PASE+ [Ravanelli et al., 2020] is an improved version of the Problem-Agnostic Speech Encoder (PASE) [Pascual et al., 2019]. It is composed of two main blocks: the speech encoder and the workers. The idea behind PASE+ is to obtain speech features using a deep neural network, the encoder, and then use these representations to solve simple tasks using shallow networks, the workers (Fig. 2.2.2). The authors trained it with 50 hours of the Librispeech corpus [Panayotov et al., 2015], and they added audio perturbations at the input, to make it more robust at inference time.

The system works directly from the raw waveform, thanks to a *SincNet*, which is a module similar to a convolutional layer, but using bandpass filters [Ravanelli and Bengio, 2019]. Next, there is a stack of 1-dimensional convolutional layers with residual connections. Before adding up these connections, a QRNN is applied, that is a type of network which can reproduce the RNN performance avoiding their high training and inference time caused by recurrence [Bradbury et al., 2016]. The block is closed by a layer which maps the output number of channels to 256.

Once the speech encoder has done its work, twelve workers use the speech representation to solve their task. The majority of these modules consist of a feed-forward layer with just one hidden layer, and they solve regression tasks, such as recovering the original waveform or the MFCC, and binary tasks, like determining if a feature vector belongs to the same audio.

Unlike the other two speech encoders we explored, the Autoregressive Predictive Coding method [Chung et al., 2019] works on the log-Mel spectrogram instead of the raw waveform. Its self-supervision objective is to predict samples of the spectrogram at n steps in the future. The authors trained it with 360 hours of Librispeech and used multiple n steps. It is made up of a stack of three GRU that acts as an encoder, and a single 1-dimensional convolutional layer which recovers the n-steps shifted log-Mel spectrogram (Fig. 2.2.3).

The last speech feature extractor we used, Wav2Vec [Schneider et al., 2019], has a slightly different structure from the others (Fig. 2.2.4). This system consists of an encoder network, which uses causal convolutional networks to obtain representations of 30ms of audio, and

a context network, that uses the same type of layers, reaching a receptive field of 810ms. The system is trained with a contrastive loss, which, using a context representation, tries to distinguish a future encoder representation from a set of fake ones. The whole Librispeech corpus was used to train this system, comprising 960 hours of speech.

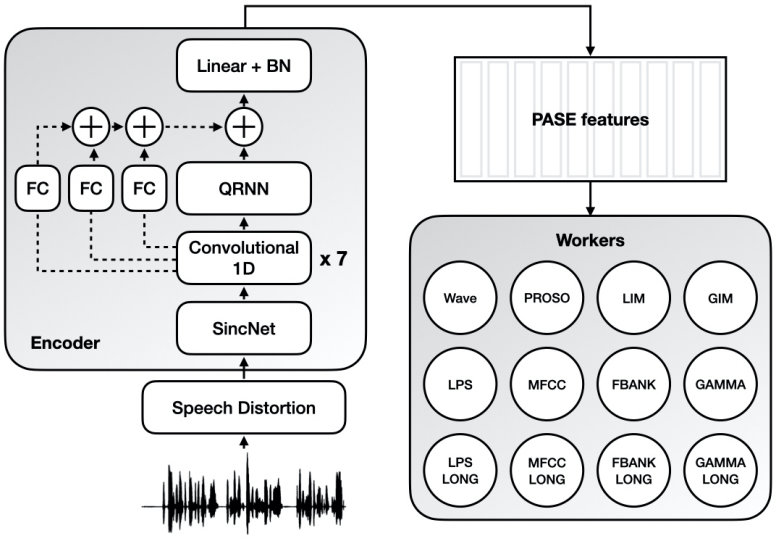


Figure 2.2.2: PASE+ architecture.

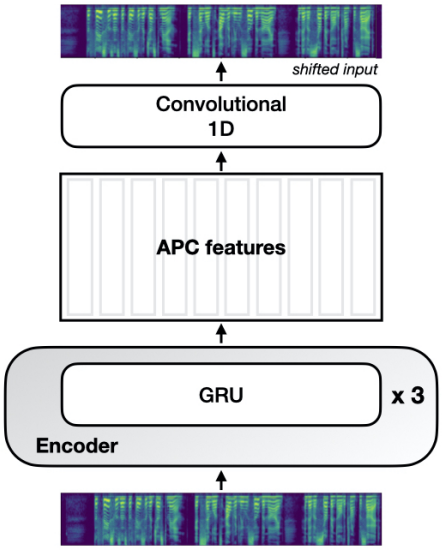


Figure 2.2.3: APC architecture.

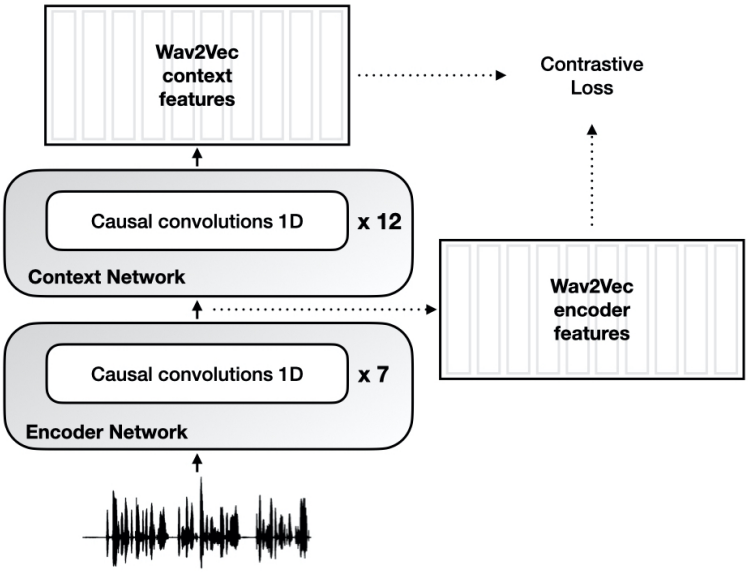


Figure 2.2.4: Wav2Vec architecture.

Chapter 3

State of the art

Nowadays, End-to-end Speech Translation is one of the most active fields of deep learning. There are plenty of researchers putting a lot of effort to surpass the best results of the cascaded approach. Still, they always fight against the fact that there are much fewer data available to train end-to-end systems than for the ASR and MT models that conform pipeline ST systems. Hereafter, we describe models that scientists have built to reach a similar performance than the cascaded approach, and all those methods that they have used to bypass the lack of enough data. We report the details of the most popular corpora for Speech Translation.

3.1 End-to-end Speech Translation

The cascade approach has some limitations, as explained in §1, that led the researchers to try to tackle Speech Translation with an end-to-end approach, i.e. without intermediate transcription representations. In this section, we describe the first steps of the end-to-end paradigm in ST, and the main advances in the field to bypass the limitations of this approach compared to the cascade, which is still predominant.

The first time the task of pairing speech with translated text was proposed was in [Duong et al., 2016], but they did not face it as a translation task per se, but rather focused on the alignments. [Bérard et al., 2016] proposed the end-to-end ST approach with an encoder-decoder system inspired by LAS [Chan et al., 2015]. However, they used a completely synthetic French-English corpus they created augmenting an MT dataset with Text-to-Speech. After that, [Bansal et al., 2017] trained for the first time an ST system with a real voice corpus, Fisher&Callhome [Post et al., 2013]. Still, they applied an unsupervised method to cluster repeated audio patterns and then train a bag of words model, which differs from the encoder-decoder approach.

The next steps have been very influential in later research in the field. [Weiss et al., 2017] introduced a multi-task architecture in which ASR and MT were trained as auxiliary tasks, sharing some modules with the primary ST system. On the other side, [Bérard et al., 2018]

introduced another successful idea, using a pre-trained encoder and decoder from ASR and MT systems, respectively. It was also the first time an end-to-end system was trained with more than 100 hours of data, with the Augmented Librispeech corpus [Kocabiyikoglu et al., 2018].

These proposals used RNN models like bidirectional LSTMs [Hochreiter and Schmidhuber, 1997]. Still, the revolutionary invention of the Transformer [Vaswani et al., 2017] supposed a change in the way sequence-to-sequence tasks were tackled. It has also influenced end-to-end ST, and [Cros Vila et al., 2018] was the first to propose a Transformer for this task.

Many researchers have used convolutional layers to reduce the size of the sequence of audio features before the Transformer encoder. Concretely, [Di Gangi et al., 2019b] presented a specific adaptation of the Transformer (Fig. 3.1.1), which also uses *2D self-attention* [Dong et al., 2018], *residual connections* and a *distance penalty* in the Transformer layers to attend near samples [Sperber et al., 2018]. Other promising modifications of the Transformers have been proposed, like [Pham et al., 2020], that adapted the use of *relative positional embeddings* to speech [Shaw et al., 2018].

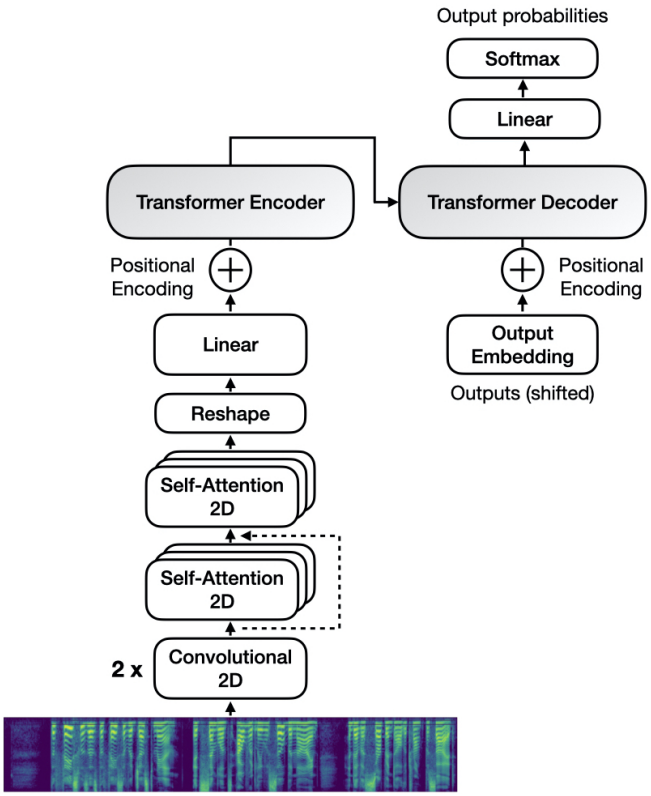


Figure 3.1.1: Architecture of the S-Transformer.

During the last years, multiple researchers proposed new systems to alleviate the lack of data for training strong enough end-to-end models, and some of them are modifications from the basic ideas we have already mentioned. In the case of multi-tasking, [Anastasopoulos and Chiang, 2018] built a triangular multi-task in which the ST decoder also receives information

from the ASR decoder. Apart from that, [Bahar et al., 2019a] proposed the use of a CTC loss after the encoder to predict transcriptions. Meanwhile, [Bansal et al., 2019] performed an in-depth study of pre-training the encoder, and concluded that the pre-training supposes considerable improvements even if the ASR language differs from both the source and target ST languages. Later, [Stoian et al., 2020] found that models with better WER in the pre-training encode more language-universal phonetic information in the later layers.

Other techniques that take advantage from ASR and MT have also been proposed. [Kano et al., 2018] introduced a curriculum learning [Bengio et al., 2009] system which started training an ASR task and progressively changed to ST. [Liu et al., 2019] built an ST system obtained with knowledge distillation from an MT model [Hinton et al., 2015]. And [Indurthi et al., 2020] used meta-learning to transfer knowledge from source tasks, i.e. ASR and MT, to target ST task [Finn et al., 2017].

Although the current availability of more abundant corpora, the use of synthetic data is still a common practice, applying a TTS on an MT corpus, or an MT system on an ASR dataset. It is the approach followed by [Jia et al., 2018], that fine-tuned an ST model with synthetic data. Additionally, [Pino et al., 2019] studied this approach profoundly and concluded that using data from TTS could be harmful in large quantities. Furthermore, the use of pseudo-labels obtained from unlabelled audio and an auxiliary ST system was also investigated by [Pino et al., 2020].

Instead of creating synthetic data, others use data augmentation techniques, that randomly modify the network inputs. In ASR, the most important is SpecAugment [Park et al., 2019], and [Bahar et al., 2019b] demonstrated that it is also useful for ST.

Another line of research focused in translating speech into multiple languages. The approaches consisted of biasing the input of the encoder [Di Gangi et al., 2019c] or the decoder [Inaguma et al., 2019] with language tokens, and they obtained improvements over one-to-one translations.

The systems we described use audio features as input, e.g. log-Mel spectrogram or MFCC. However, other researchers suggested a different line of research and, instead of working with standard speech features, proposed to work with phoneme-like representations, to reduce the sequence length, improving results and reducing the training time [Salesky et al., 2019][Salesky and Black, 2020]. Recently, other scientists proposed the use of self-supervised speech representations obtained with wav2vec [Schneider et al., 2019] and vq-wav2vec [Baevski et al., 2019], but, unlike in our project, using LSTM models [Wu et al., 2020].

The growth of the end-to-end approach has been represented clearly in the evaluation campaigns of the IWSLT, that have an independent category for end-to-end ST since the 2018 edition. These campaigns, which evaluate with a test set of English-German translations from TED talks, allow accurate tracking of the differences between cascade and end-to-end ST systems through the years. As can be seen in table 3.1.1, the end-to-end approach has im-

proved in every edition, reaching out the cascade system this year [Potapczyk and Przybysz, 2020], although the best global result is still from a pipeline approach. There has also been an increase in the number of participants for the end-to-end category, being four participants at 2018 [Turchi et al., 2018], five the next year [Niehues et al., 2019], and eight in the 2020 edition [Ansari et al., 2020]; which denotes a growing interest in this approach by the research community.

	Cascade		**End-to-end**	
	Team	BLEU (↑)	Team	BLEU (↑)
2018	*Sogou-TIIC*	**28.19**	*USTC-NEL*	18.32
2019	*KIT*	**21.55**	*SRPOL*	19.96
2020	*AppTek*	25.06	*SRPOL*	**25.30**

Table 3.1.1: Best results of the IWSLT Speech Translation evaluation campaigns (2018-20). Results on the English-German TED talks test set.

3.2 Speech Translation corpora

Collecting enough data to train deep learning models is always a challenge, and Speech Translation is not an exception. Still, investigators of this field struggle especially the lack of data, compared to others working on tasks like Automatic Speech Recognition or Machine Translation who are years ahead in it.

The standard approach to create a corpus is to find a way to translate the transcriptions of an already existing ASR dataset. Following this idea was created the Fisher and Callhome Spanish-English Speech Translation Corpus, which was the first dataset for ST [Post et al., 2013]. It was built from two Spanish ASR datasets, consisting of transcribed phone calls, which were translated to English using crowdsourcing. It was the standard in the field since the appearance of more resourced datasets.

Another ASR corpus which allowed the generation of ST corpora is Librispeech [Panayotov et al., 2015], which contains audiobooks from the LibriVox project and is very well-known. In this case, no translators are needed, since the books are already translated into multiple languages, instead of that, automatic multilingual text alignment techniques can be used. Two datasets were created following this approach, Augmented Librispeech, that is an English-French dataset [Kocabiyikoglu et al., 2018], and LibriVoxDeEn which did the same but with German audiobooks translated into English [Beilharz et al., 2020].

Some researchers also saw a great opportunity in videos available publicly on the internet, that sometimes contain subtitles as well. The creators of How2 followed this approach, building a multimodal dataset with paired video, speech and English subtitles extracted from Youtube, and translated to Portuguese with crowdsourcing methods [Sanabria et al., 2018].

However, others have found sources with already translated subtitles. It is the case of Europarl-ST, that was obtained from the European Parliament debates, and contains audio-text pairs between 6 languages in all translation directions [Iranzo-Sánchez et al., 2019]. On the other side, MuST-C, which was derived from TED talks [Di Gangi et al., 2019a], includes the largest language pairs available nowadays (Fig. 3.2.1) .

Finally, another line of research has focused on using the resources available from the Common Voice project[1], which collects speech samples from volunteers around the world. In this context were created CoVoST, that offers translations from 11 languages into English [Wang et al., 2020a], and CoVoST 2, which supposed a massive update, providing translations from 21 languages into English, and from English into 15 languages [Wang et al., 2020b].

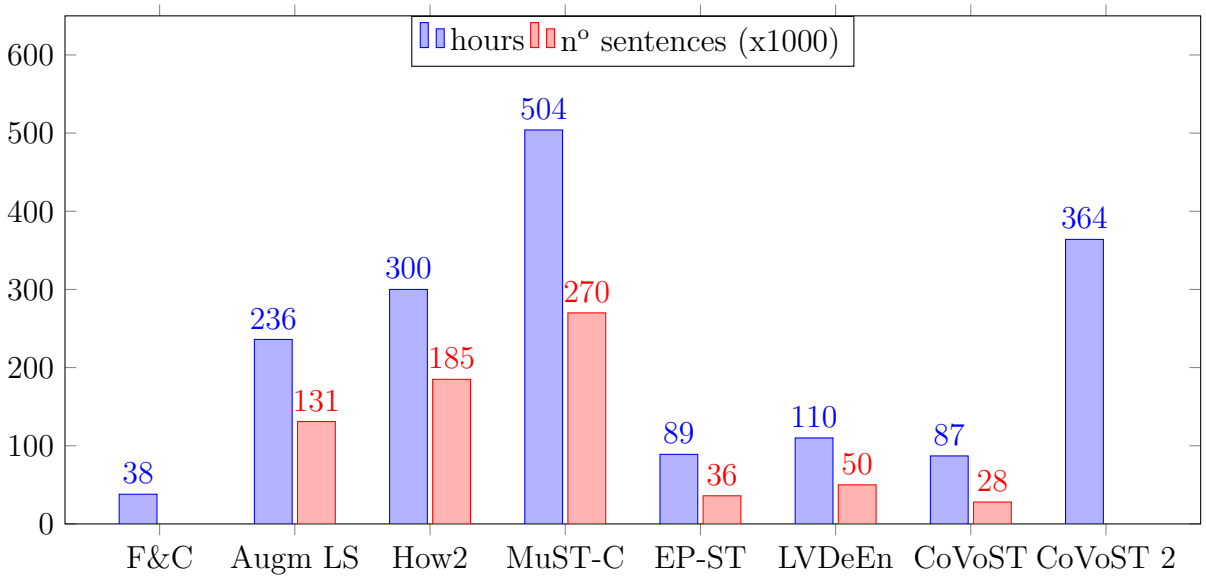


Figure 3.2.1: ST corpora sizes. Training data in the largest language pair of each corpus[2].

[1]https://commonvoice.mozilla.org
[2]The Fisher&Callhome and CoVoST 2 papers do not specify the number of sentences.

Chapter 4

Methodology

In this project, we trained four different End-to-end Speech Translation models, including three variations of the system we propose, and the S-Transformer as a baseline [Di Gangi et al., 2019b]. Moreover, we built a cascade system using pre-trained ASR and MT models, to contrast it with the end-to-end systems. From now on, we describe all these models, giving details of their architectures.

We started training our baseline model, the S-Transformer. This model uses log-Mel spectrograms, and transforms them with convolutional and self-attentional layers attached before the Transformer encoder (Fig. 3.1.1).

The first two 2D convolutional layers capture patterns in the localities of the spectrograms and, thanks to the stride, they reduce the feature maps by four in both dimensions. It is especially crucial because decreasing the sequence length avoids memory issues. The two 2D self-attention layers, which were introduced by [Dong et al., 2018] are in charge of modelling the long-range dependencies of the spectrogram that convolutional layers cannot capture. This system also adds a logarithmic distance penalty to the self-attention layers of the Transformer encoder, which biases these layers towards the local context [Sperber et al., 2018].

We pre-trained it for Automatic Speech Recognition, which is a common practice in the field, because, as many researchers suggest, using an encoder trained for ASR improves the performance of the final ST system, since the first is an easier task (see §3.1).

The system we propose is a novel architecture which takes advantage from pre-trained speech encoders, in conjunction with a Transformer model. Concretely, we explored the use of PASE+, APC and Wav2Vec as speech encoders, instead of using classical speech features, like the log-Mel spectrogram (see §2.2). However, feeding the Transformer model directly with the speech representations would cause out-of-memory errors.

Hence, we designed the *Coupling network*, which picks some key concepts from the S-Transformer. It consists of a stack of two 1D time-channel separable convolutional layers,

which reduce the sequence length by four, finding local patterns only in the time dimension. Each of these layers also has a ReLU activation and batch normalization. This design (Fig. 4.0.1) has not been the only one we tested, and in Appendix A we discuss the different approaches we explored. Moreover, we also applied the logarithmic distance penalty in the Transformer encoder.

In addition to the ASR pre-training and the ST training, we added a third step, in which we tried to get more out of the model by fine-tuning the speech encoder, which was previously frozen, while freezing the decoder.

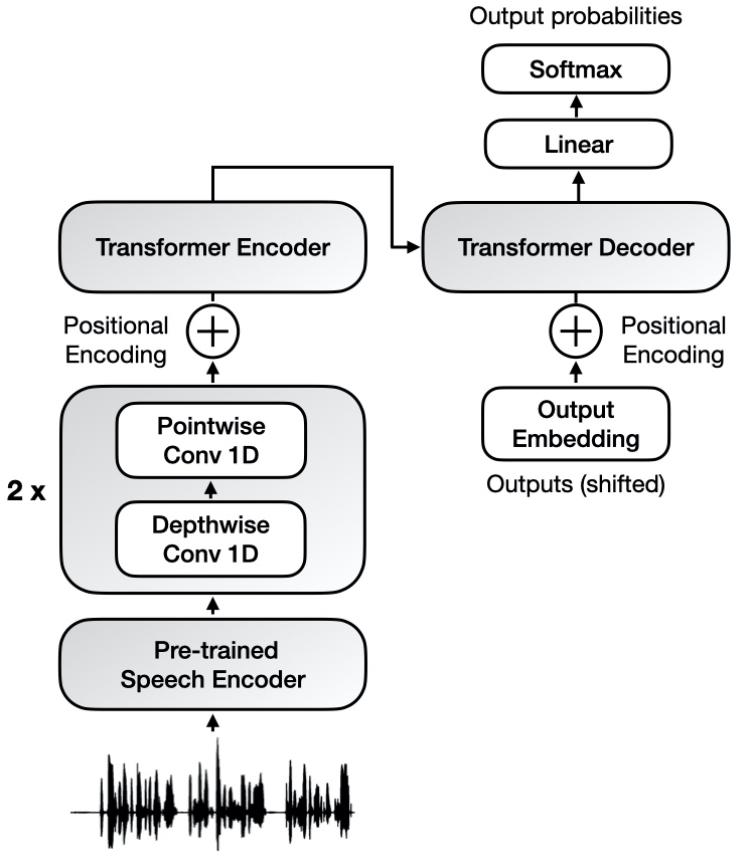


Figure 4.0.1: Architecture of our model.

Finally, we also built a baseline cascade system, using pre-trained ASR and MT models, to contrast the end-to-end approaches. The Speech Recognition model we used is QuartzNet [Kriman et al., 2019], a system developed by NVIDIA, made up of multiple blocks containing time-channel separable convolutions. It was trained with data from Librispeech [Panayotov et al., 2015], Common Voice, Fisher [Cieri et al., 2004], WSJ [Paul and Baker, 1992] and Switchboard [Godfrey et al., 1992]. Regarding the MT system, we used a Transformer model from the OpusMT project [Tiedemann and Thottingal, 2020], a collection of over 1,000 pre-trained translation models, developed by the Language Technology Research Group at the University of Helsinki. We executed this model through the Hugging Face transformers library [Wolf et al., 2019].

Chapter 5

Experimental framework

In this chapter, we describe how we executed our experiments. First, we explain the process of choosing a corpus, its preparation and the preprocessing we applied. Second, we give details of how we implemented our Speech Translation system, and, finally, we detail the configuration we used in our experiments.

5.1 Dataset

In this section, we explain the datasets we used in our project, as well as each step we had to perform on them before executing any experiment. First, we will discuss the corpora we chose to train our systems, and then, we describe the work done for preparing the datasets, in which we spent some time on the structure standardization. Finally, we explain how we tokenized and encoded text data, as well as the dictionary building.

Our primary requisite to choose a corpus was that it was a standard for training ST systems, to be able to compare our results with others. But we also needed a large enough dataset to train a big model like ours.

Most of the early end-to-end ST studies used the Fisher and Callhome Spanish-English Speech Translation Corpus for training their systems [Duong et al., 2016][Weiss et al., 2017][Anastasopoulos and Chiang, 2018][Cros Vila et al., 2018]. However, this dataset was too small, containing only 38 hours of speech, approximately.

Hence, we decided to begin our work using the Augmented Librispeech corpus, an English-French ST dataset based on a well known ASR corpus, as explained in §3.2. It almost triplicates the amount of data available in the previous option, only with the clean data split[1], and some papers that introduce novel techniques used it [Bérard et al., 2018][Liu et al., 2019][Inaguma et al., 2019][Pino et al., 2019].

[1]There are up to 236 hours available for training (Fig. 3.2.1)

However, during the project, MuST-C started gaining popularity [Indurthi et al., 2020], since it is the corpus with the largest data splits between English and 8 of the most important European languages, with sizes from 385 to 504 hours. Moreover, the authors of the model we used as a baseline trained it with this corpus [Di Gangi et al., 2019b]; consequently, we finally decided to work with the MuST-C corpus instead of the Augmented Librispeech. Still, we decided to keep working the same language pair, English-French.

During the development of our project, we realised we were spending too much time with data preparation, adapting the code to the different data structure of each corpus. Furthermore, we cannot keep unused datasets in our server, since the shared storage is limited; hence we may have to download and prepare the same corpus multiple times along time.

We thought we needed a way to save those data preparation steps and to standardize corpora folder structure, to convert any dataset into a particular format, in such a way that it fits the later preprocessing pipeline seamlessly. Therefore, we created corpora-recipes[2], a simple library that performs this job for speech-to-text datasets and that, at this moment, is prepared to work with Librispeech, Augmented Librispeech and MuST-C. Still, we plan to grow it in the future.

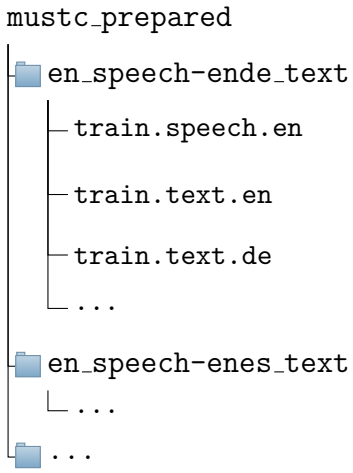


Figure 5.1.1: Corpus folder structure.

The standardized folder structure consists of a directory comprising subdirectories for each split of paired data of the dataset (Fig. 5.1.1). Each subdirectory includes the files with a simple name that gives information about the subset, the data type and the language. It is important to remark that the new folder does not include audio data, the speech files only contain references to the original ones, as well as speaker information, duration of each utterance and an offset. On the other hand, text files contain the sentences aligned with the speech references.

The preprocessing steps we performed and the code we developed to make our experiments are both based on this directory structure. Using this library made it easier for us to change

[2]https://github.com/gegallego/corpora-recipes

the dataset efficiently, and will save us time in the future when training our system with new corpora and language pairs.

We want to make it clear that we did not perform any audio data preprocessing because, unlike other speech-to-text systems that extract speech features beforehand, ours use raw audio at the input. The reason for implementing it in this way is that we compared various speech encoders, and some of them work directly from the waveform, instead of other typical strategies like using the log-Mel spectrogram (see §5.2).

We preprocessed the text data because it is necessary to transform it into a numerical format, to feed the model. Moreover, it is also important for other purposes, like removing inconsistencies. We began the preprocessing by normalizing the punctuation, which standardize the quotation style and other language-dependent symbols, and removes unnecessary spaces around punctuation[3]. After that, we tokenized the sentences[3] to make it easier for the text encoder, and we de-escaped the special characters.

```
« C'est un prisonnier», se dit Edmond avec une indicible joie.
                              ↓
" C' est un prisonnier " , se dit Edmond avec une indicible joie .
```

In the case of the data corresponding to the speech transcriptions, that is to say, the English text, we added some steps in the preprocessing. It is a common approach to train ASR systems in a case-insensitive manner [Chan et al., 2015] and without punctuation. Hence, we lower-cased and removed punctuation from the already tokenized files.

```
In fact, it's a new invention of the wheel.
                    ↓
 in fact it s a new invention of the wheel
```

The next step was to build a dictionary to encode the text. The objective is to create a fixed-length vocabulary with an index for each element, and then substitute text by numerical data. Classical MT used vocabularies at a word-level or character-level, but other options gained popularity, like Byte-Pair Encoding [Sennrich et al., 2016]. Following the trend in MT of using the latter option, we decided to build our dictionary of translation data with BPE and a size of 8000 sub-words.

However, although ASR research has to face text encoding too, the practices are quite different, since it is more closely related to interpreting the human voice at each instant. For this reason, we used a character-level encoding for the transcription data [Collobert et al., 2016].

5.2 Implementation

One of the objectives we set for this project was to have a reliable and flexible code to ease launching new experiments in the future. Here, we give details of the implementation of our Speech Translation system, using the Fairseq framework.

[3]We used Moses: https://github.com/moses-smt/mosesdecoder

Fairseq is a sequence-to-sequence toolkit from Facebook AI Research [Ott et al., 2019]. It reduces the initial effort that researchers need to do to start launching their experiments; since it already contains multiple modules, training scripts and other useful tools. Nevertheless, it is mainly focused on Machine Translation and other Natural Language Processing tasks, and it does not contain many modules dedicated to speech yet.

We considered using the code from the FBK-Fairseq-ST repository[4], which is focused on Speech Translation and is maintained by the creators of the S-Transformer [Di Gangi et al., 2019b]. However, our implementation had some special needs that made this library not suitable for us: we wanted to load directly raw audio into the model, to be able to process it with the speech encoders.

This fact, and the need for working with modular and flexible code, to ease our future work, encouraged us to create a new ST library from scratch, implementing the Fairseq structure. We desire that this repository[5] becomes a shared resource for everyone in our group working in ST.

Typically, speech samples are pre-processed, extracting features like the log-Mel spectrogram or the MFCC, and then, the code directly loads these pre-computed representations into the model. Nevertheless, the way we conceived our model implementation is quite different from that. We decided that the speech encoder, whichever it was, had to be inside of the model, to be able to fine-tune it. We also wanted to keep the possibility of using classical speech features instead of these encoders, so we also implemented them by using the torchaudio package, that can compute these features even if the tensors are on a GPU. The result is a model which loads raw audio waveforms and in which the user can seamlessly change the speech feature extractor to be used.

We also implemented a data augmentation strategy called SpecAugment [Park et al., 2019], which randomly masks blocks of the speech representations during training. However, we did not use it in the final training, to make a fair comparison between our system and the baseline, which does not use it.

5.3 Parameters

To ease the reproducibility of our experiments, in this section, we detail the configuration we used to train our models. Notice that we do not specify the parameters for training the S-Transformer baseline model, because we used the same ones the author specifies in the FBK-Fairseq-ST repository[4]. Apart from that, we did not need any configuration for the cascade system we built, since we got its results with the pre-trained ASR and MT models, without any modification.

[4]https://github.com/mattiadg/FBK-Fairseq-ST
[5]https://github.com/gegallego/end2end-st

Regarding the speech encoders, we used the pre-trained models available on their official repositories[6][7][8]. Although in [Chung et al., 2019] two APC models are described, one based on a GRU and another based on a Transformer; only the first one is available in their repository[7]. There are multiple GRU-based APC models, which vary the step size used during the self-supervision training (§2.2). We decided to use the one with $n = 3$, since it is the one with the best results reported in [Chung and Glass, 2020]. On the other side, we only used the Speech network of the Wav2Vec model, and we get rid of the Context network (Fig. 2.2.4).

As we described in §4, the Coupling network consists of two layers of 1D Time-Channel separable convolutions. These layers use a kernel size of 3, with stride 2 and padding 1; they keep the same number of channels of the speech representation in the first layer (i.e. 256 for PASE+ and 512 for the others), and they convert it to the embedding size of the Transformer in the second one. After each Time-Channel separable convolution layer, there is a ReLU activation, a batch normalization layer, and dropout.

We built our model with 6 layers in the Transformer encoder and decoder, with an embedding size of 512. The feed-forward network from each layer has a hidden size of 1024, and we used 8 self-attention heads.

We trained the system with an Adam optimizer [Kingma and Ba, 2014], using a learning rate of $5 \cdot 10^{-3}$, and an inverse square root scheduler. Moreover, we applied a warm-up for the first 4000 updates, in which the learning rate grows linearly from $3 \cdot 10^{-4}$. During the fine-tuning stage, we used a constant learning rate of $1 \cdot 10^{-3}$. The loss we used is the Cross-entropy, with a label smoothing factor of 0.1. We also applied a dropout of 0.1, and we clipped the gradient to 20, to avoid exploding gradients [Pascanu et al., 2012]. During training and validation, we used teacher forcing [Williams and Zipser, 1989], but the generation of the test results was done using beam search [Graves, 2012], with a beam size of 5.

We used an NVIDIA GeForce RTX 2080 Ti GPU for training, but setting an update frequency of 64, which simulates the use of 64 GPUs. Moreover, we loaded 8 sentences per update, which supposes an effective batch size of 512. Furthermore, during training and validation, we removed utterances longer than 14 seconds and sentences with more than 300 tokens, to avoid memory problems.

[6]https://github.com/santi-pdp/pase
[7]https://github.com/iamyuanchung/Autoregressive-Predictive-Coding
[8]https://github.com/pytorch/fairseq/tree/master/examples/wav2vec

Chapter 6

Results and discussion

In this chapter, we present the results of the experiments we carried out. We start by exposing them objectively (Table 6.0.1), and then, we discuss the possible reasons behind the differences in the performance obtained with each approach.

	ASR (WER ↓)	**ST (BLEU ↑)**	**ST$^{\diamondsuit}$ (BLEU ↑)**
Cascade	**20.69**	21.1	-
S-Transformer	23.65	**27.0**	-
Ours (PASE+)	36.80	18.0	20.5
Ours (APC)	32.92	14.4	14.6
Ours (Wav2Vec)	27.67	20.0	22.1

Table 6.0.1: ASR & ST results.
$\diamondsuit$: With fine-tuning.

The best result in the Automatic Speech Recognition task corresponds to the pre-trained QuartzNet [Kriman et al., 2019] we used in the cascade system. Still, note that this result is much worse than the reported for the Librispeech test-clean subset (3.90%). On the other hand, the S-Transformer has the best performance in Speech Translation, followed by the fine-tuned Wav2Vec model, both surpassing the cascade approach.

The model using Wav2Vec is the best we proposed, getting 20.0 BLEU even without fine-tuning, and improving up to 22.1 BLEU when it does. Although PASE+ has the worst results in ASR, it has much better results than APC in ST. The improvement of the models when fine-tuning the speech encoder is notable in PASE+ (+2.5 BLEU) and Wav2Vec (+2.1 BLEU), but not significant for APC (+0.2 BLEU).

The first conclusion we extract from the results is that we improved the cascade approach with the S-Transformer and the model we proposed using Wav2Vec. However, we must take into account some details which might be causing that the cascade approach falls behind the end-to-end. The MuST-C corpus contains speech data from a particular domain, live presentations from TED Talks, for which the QuartzNet has neven been trained; what explains the difference in the performance of this system with respect to the one reported for Librispeech. Moreover, the ASR module has more influence in the final results of the cascade ST than in the end-to-end model, as the first one suffers from error propagation, and the latter only uses the ASR for pre-training purposes. It could explain that, despite being the best ASR, it is probably not good enough to get competitive results in cascade ST.

Additionally, we detected another limitation of the cascade systems. Typically, ASR models are trained with uncased text and without punctuation, but ST systems are evaluated with casing and punctuation. Hence, we are giving the responsibility to regenerate these elements to the MT module, which is impossible when using a pre-trained model like in this case. Using a cascade system with pre-trained components did not result as a good option as we expected; and we conclude that the choice of the ASR and MT modules is crucial to get a good performance from this type of systems.

The same out-of-domain factor that is negatively affecting the cascade approach could also be harming the systems we propose, as the three pre-trained speech encoders we used were all trained on Librispeech. Moreover, it explains why the performance increases so much when we fine-tune the speech encoders: because we are adapting them to the new domain.

Furthermore, taking into account that we used the same Transformer configuration than the S-Transformer's, we are considering that perhaps the Coupling network is causing that our systems fall behind. We think that this component might not be capable of compressing the speech representations properly, and that we might not be feeding the Transformer Encoder with rich enough features. Moreover, in the case of the model using APC, which obtained the worst results in ST, we hypothesise that the Coupling network could not be appropriate for the architecture of its speech encoder at all. The outputs of APC come from the hidden states of a GRU layer, and we think that the locality concept in the temporal dimension that we exploit in the Coupling network, is not applicable for the outputs of an RNN.

We were gratefully surprised by the improvements obtained when unfreezing the speech encoder and freezing the Transformer decoder. We believe that it opens the door to further investigation in this direction. However, in the case of APC, we did not get a significant improvement, which we think that confirms our idea that the Coupling network is not adequate for this speech encoder.

Finally, we also compared the training speed of each of our models. We conclude that PASE+ is the fastest implementation, while Wav2Vec requires a +11.7% more time, and APC +28.8%. The models using PASE+ and Wav2Vec are faster because they are mainly composed of convolutions, whereas APC is based on recurrence, which is more inefficient.

Chapter 7

Conclusions and future work

End-to-end Speech Translation is one of the most exciting research fields in deep learning nowadays. Lots of researchers are working together to improve this type of systems and to make it possible to use them in real-world applications soon.

This project contributes to the consolidation of ST as a line of research in the UPC Machine Translation group. For this reason, we have invested time in performing an in-depth study of the state-of-the-art and catching up with the main trends of the field. Now, we have a comprehensive view that allows us to decide in which direction we want to develop our future research.

In this context, another goal of the project was to develop tools to facilitate the launching of new experiments in the future. We decided to create a new repository with tools for Speech Translation, which we will make publicly available from the Machine Translation group's Github, and we will keep updated with the new modules we will create in our next steps in the field. Moreover, we have also created an additional tool, which standardises the preparation of speech-to-text corpora, allowing us to work with different datasets without wasting too much time in data preparation. We put efforts to build these tools, because we have always focused on the long term, intending to build a solid foundation for our future work.

The systems we proposed do not reach the results of the baseline, but they open multiple doors to further research on the use of pre-trained speech encoders for ST. Why is there such a difference between the performance of the different options we explored? We will need to study more in-depth the characteristics that make some of them work better for this task than others, and perhaps we will need to redesign our architecture to exploit better each of the speech representations we use. Furthermore, the substantial improvement we obtained when fine-tuning the speech encoders demonstrates that the decision to include the feature extraction step inside the model was correct, and we hypothesise that it will be the key to compete with state-of-the-art ST systems in the future.

We have gathered lots of ideas to keep working in ST during the PhD I am about to start. First, as said before, we will analyse in-depth what caused such a different performance between the speech encoders we used. Second, we will also keep trying variations of the current model, like modifying the number of layers of the Transformer or exploring the use of other strategies to couple the speech encoders. Third, we would like to implement some popular techniques, like multitask learning, knowledge distillation or using TTS as data augmentation. Finally, we will explore how to reduce the influence of the domain and the speaker in the performance of these systems.

Bibliography

Antonios Anastasopoulos and David Chiang. Tied Multitask Learning for Neural Speech Translation. pages 82–91, 2018. doi: 10.18653/v1/n18-1008.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alex Waibel, and Changhan Wang. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. Technical report, 2020.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. pages 1–12, 10 2019. URL `http://arxiv.org/abs/1910.05453`.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. A Comparative Study on End-to-end Speech to Text Translation. 11 2019a. URL `http://arxiv.org/abs/1911.08870`.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. On Using SpecAugment for End-to-End Speech Translation. 11 2019b. URL `http://arxiv.org/abs/1911.08876`.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. Towards speech-to-text translation without speech recognition. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, 2:474–479, 2017. doi: 10.18653/v1/e17-2076.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North*, pages 58–68, Stroudsburg, PA, USA, 9 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1006. URL `http://aclweb.org/anthology/N19-1006http://arxiv.org/abs/1809.01431`.

Benjamin Beilharz, Xin Sun, Sariya Karimova, and Stefan Riezler. LibriVoxDeEn: A Corpus for German-to-English Speech Translation and German Speech Recognition. pages 11–16, 2020. doi: 10.1016/j.artint. URL `https://dx.doi.org/10.1016/j.artint.`

Yoshua Bengio, umontrealca Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum Learning. Technical report, 2009.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. (Nips), 2016. URL http://arxiv.org/abs/1612.01744.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-End Automatic Speech Translation of Audiobooks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:6224–6228, 2018. ISSN 15206149. doi: 10.1109/ICASSP.2018.8461690.

Laurent Besacier, Bowen Zhou, and Yuqing Gao. TOWARDS SPEECH TRANSLATION OF NON WRITTEN LANGUAGES. Technical report, 2006.

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-Recurrent Neural Networks. 11 2016. URL http://arxiv.org/abs/1611.01576.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. 2020. URL http://arxiv.org/abs/2005.14165.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, Attend and Spell. pages 1–16, 8 2015. URL http://arxiv.org/abs/1508.01211.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 6 2014. URL http://arxiv.org/abs/1406.1078.

Yu-an Chung and James Glass. Generative Pre-Training for Speech with Autoregressive Predictive Coding. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, number 1, pages 3497–3501. IEEE, 5 2020. ISBN 978-1-5090-6631-5. doi: 10.1109/ICASSP40776.2020.9054438. URL http://arxiv.org/abs/1910.12607https://ieeexplore.ieee.org/document/9054438/.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. An Unsupervised Autoregressive Model for Speech Representation Learning. 4 2019. URL http://arxiv.org/abs/1904.03240.

Christopher Cieri, David Miller, and Kevin Walker. The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, 2004. European Lan-

guage Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/767.pdf.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. pages 1–8, 9 2016. URL http://arxiv.org/abs/1609.03193.

Laura Cros Vila, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-Jussà. End-to-End Speech Translation with the Transformer. In *IberSPEECH 2018*, number November, pages 60–63, ISCA, 11 2018. ISCA. doi: 10.21437/IberSPEECH.2018-13. URL http://www.isca-speech.org/archive/IberSPEECH_2018/abstracts/IberS18_P1-9_Cross-Vila.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (Mlm), 10 2018. URL http://arxiv.org/abs/1810.04805.

Mattia. A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. MuST-C: a Multilingual Speech Translation Corpus Mattia. *Naacl*, (2018):2012–2017, 2019a. URL https://www.aclweb.org/anthology/N19-1202.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. Adapting Transformer to End-to-End Spoken Language Translation. In *Interspeech 2019*, volume 2019-Septe, pages 1133–1137, ISCA, 9 2019b. ISCA. doi: 10.21437/Interspeech.2019-3045. URL http://www.isca-speech.org/archive/Interspeech_2019/abstracts/3045.html.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. One-to-Many Multilingual End-to-End Speech Translation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 585–592. IEEE, 12 2019c. ISBN 978-1-7281-0306-8. doi: 10.1109/ASRU46091.2019.9004003. URL http://arxiv.org/abs/1910.03320http://arxiv.org/abs/1910.03320.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. Technical report, 2015.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2018-April, pages 5884–5888. IEEE, 4 2018. ISBN 978-1-5386-4658-8. doi: 10.1109/ICASSP.2018.8462506. URL https://ieeexplore.ieee.org/document/8462506/.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. An attentional model for speech translation without transcription. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, (i):949–959, 2016. doi: 10.18653/v1/n16-1109.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. Technical report, 2017.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. 5 2017. URL http://arxiv.org/abs/1705.03122.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

Alex Graves. Sequence Transduction with Recurrent Neural Networks. 11 2012. URL http://arxiv.org/abs/1211.3711.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. 3 2015. URL http://arxiv.org/abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. Multilingual End-to-End Speech Translation. 10 2019. URL http://arxiv.org/abs/1910.03320http://arxiv.org/abs/1910.00254.

Sathish Indurthi, Houjeung Han, Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. END-END SPEECH-TO-TEXT TRANSLATION WITH MODALITY AGNOSTIC META-LEARNING. 2020. ISBN 9781509066315. URL http://www.statmt.org/wmt16/.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. 11 2019. URL http://arxiv.org/abs/1911.03167.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation. 11 2018. URL http://arxiv.org/abs/1811.02050.

Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. Structured-based Curriculum Learning for End-to-end English-Japanese Speech Translation. 2 2018. URL http://arxiv.org/abs/1802.06003.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014. URL http://arxiv.org/abs/1412.6980.

Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. Augmenting Librispeech with French translations: A multimodal corpus for direct speech translation evaluation. *LREC*

2018 - 11th International Conference on Language Resources and Evaluation, pages 1–5, 2018.

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. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6124–6128, 10 2019. URL http://arxiv.org/abs/1910.10261.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, Patrice Castonguay, Mariya Popova, Jocelyn Huang, and Jonathan M. Cohen. NeMo: a toolkit for building AI applications using Neural Modules. 9 2019. URL http://arxiv.org/abs/1909.09577.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-end speech translation with knowledge distillation. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019-Septe:1128–1132, 2019. ISSN 19909772. doi: 10.21437/Interspeech.2019-2582.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. 8 2015. URL http://arxiv.org/abs/1508.04025.

Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. 3 2016. URL http://arxiv.org/abs/1603.08561.

J Niehues,) R Cattoni,) S Stüker,) M Negri,) M Turchi,) T Ha,) E Salesky,) R Sanabria,) L Barrault,) L Specia, and) M Federico. The IWSLT 2019 Evaluation Campaign. Technical report, 2019. URL https://github.com/isl-mt/SLT.KIT.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. 4 2019. URL http://arxiv.org/abs/1904.01038.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 2015-August, pages 5206–5210. Institute of Electrical and Electronics Engineers Inc., 8 2015. ISBN 9781467369978. doi: 10.1109/ICASSP.2015.7178964.

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. 4 2019. doi: 10.21437/Interspeech.2019-2680. URL http://arxiv.org/abs/1904.08779http://dx.doi.org/10.21437/Interspeech.2019-2680.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. 2 2018. URL http://arxiv.org/abs/1802.05751.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. 11 2012. URL http://arxiv.org/abs/1211.5063.

Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019-Septe:161–165, 2019. ISSN 19909772. doi: 10.21437/Interspeech.2019-2605.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 12 2019. URL http://arxiv.org/abs/1912.01703.

Douglas B Paul and Janet M Baker. The Design for the Wall Street Journal-based CSR Corpus*. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman*, New York, 1992. URL https://www.aclweb.org/anthology/H92-1073.

Ngoc-Quan Pham, Thanh-Le Ha, Tuan-Nam Nguyen, Thai-Son Nguyen, Elizabeth Salesky, Sebastian Stueker, Jan Niehues, and Alexander Waibel. Relative Positional Encoding for Speech Recognition and Direct Translation. pages 1–5, 5 2020. URL http://arxiv.org/abs/2005.09940.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D. McCarthy, and Deepak Gopinath. Harnessing Indirect Training Data for End-to-End Automatic Speech Translation: Tricks of the Trade. 2019. URL http://arxiv.org/abs/1909.06515.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. Self-Training for End-to-End Speech Translation. (Ll), 6 2020. URL http://arxiv.org/abs/2006.02490.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus. *Proceedings of IWSLT*, 2013.

Tomasz Potapczyk and Pawel Przybysz. SRPOL's System for the IWSLT 2020 End-to-End Speech Translation Task. pages 89–94, 2020. doi: 10.18653/v1/P17. URL https://doi.org/10.18653/v1/P17.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, 2019. URL https://github.com/codelucas/newspaper.

Mirco Ravanelli and Yoshua Bengio. Speaker Recognition from Raw Waveform with SincNet. *2018 IEEE Spoken Language Technology Workshop, SLT 2018 - Proceedings*, pages 1021–1028, 2019. doi: 10.1109/SLT.2018.8639585.

Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-Task Self-Supervised Learning for Robust Speech Recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 5 2020. ISBN 978-1-5090-6631-5. doi: 10.1109/ICASSP40776.2020.9053569. URL http://arxiv.org/abs/2001.09239https://ieeexplore.ieee.org/document/9053569/.

Elizabeth Salesky and Alan W Black. Phone Features Improve Speech Translation. 5 2020. URL http://arxiv.org/abs/2005.13681.

Elizabeth Salesky, Matthias Sperber, and Alan W Black. Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Stroudsburg, PA, USA, 6 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1179. URL http://arxiv.org/abs/1906.01199https://www.aclweb.org/anthology/P19-1179.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: A Large-scale Dataset for Multimodal Language Understanding. (Nips), 2018. URL http://arxiv.org/abs/1811.00347.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Interspeech 2019*, pages 3465–3469, ISCA, 9 2019. ISCA. doi: 10.21437/Interspeech.2019-1873. URL http://www.isca-speech.org/archive/Interspeech_2019/abstracts/1873.html.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. Technical report, 2016.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. Technical report, 2018.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. Self-attentional acoustic models. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2018-September, pages 3723–3727. International Speech Communication Association, 2018. doi: 10.21437/Interspeech.2018-1910.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. *Transactions of the Association for Computational Linguistics*, 7(1):313–325, 4 2019. ISSN 2307-387X. doi: 10.1162/tacl{_}a{_}00270. URL http://arxiv.org/abs/1904.07209.

Mihaela C. Stoian, Sameer Bansal, and Sharon Goldwater. Analyzing ASR Pretraining for Low-Resource Speech-to-Text Translation. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913, 2020. doi: 10.1109/icassp40776.2020.9053847.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. 2014.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT-Building open translation services for the World. Technical report, 2020. URL `http://opus.nlpl.eu`.

Marco Turchi, Jan Niehues, and Marcello Frederico. 15th International Workshop on Spoken Language Translation. Bruges, 2018. URL `www.iwslt.org`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. (Nips), 2017. URL `http://arxiv.org/abs/1706.03762`.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus. 2020a. URL `http://arxiv.org/abs/2002.01320`.

Changhan Wang, Anne Wu, and Juan Pino. CoVoST 2: A Massively Multilingual Speech-to-Text Translation Corpus. 7 2020b. URL `http://arxiv.org/abs/2007.10310`.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-sequence models can directly translate foreign speech. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017-Augus:2625–2629, 2017. ISSN 19909772. doi: 10.21437/Interspeech.2017-503.

Ronald J Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Technical report, 1989.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. 10 2019. URL `http://arxiv.org/abs/1910.03771`.

Anne Wu, Changhan Wang, Juan Pino, and Jiatao Gu. Self-Supervised Representations Improve End-to-End Speech Translation. 6 2020. URL `http://arxiv.org/abs/2006.12124`.

Appendix A

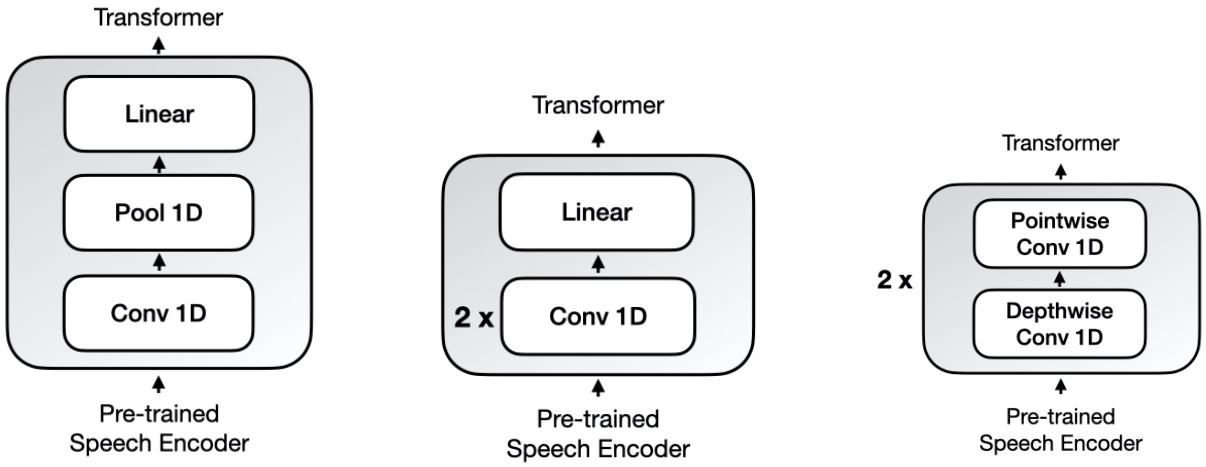
Coupling network

A naive approach for an end-to-end Speech Translation model would be to feed the Transformer encoder with speech features directly. However, taking into account that speech representations are usually computed with a step size of 10 ms[1], the sequences are too long to feed a Transformer and would cause out-of-memory errors. Hence, reducing the time dimension of speech representations is mandatory and, for this reason, we designed the Coupling network.

Our original idea, and the one we had in mind for a long time during the project, consisted of using the S-Transformer [Di Gangi et al., 2019b] as is, but substituting the log-Mel spectrogram by a pre-trained speech feature extractor. However, we did not manage to train the system with this approach, and we started questioning ourselves the architecture. We realised that the layers used before the Transformer were designed taking into account the characteristics of a log-Mel spectrogram, but it might not fit our speech representations. The goal of the 2D convolutions is to capture local patterns in the two dimensions, which is applicable in a spectrogram, where both the time and the frequency axes have a sense of locality. However, we realised that the speech representations we were using do not have this characteristic, and we hypothesised that they only have a sense of locality in the time dimension. Furthermore, we also examined the need of the 2D self-attention layers, whose objective is to capture the long-range context. We decided that we were using powerful enough speech encoders to get rid of these layers, and especially taking into account that the sequence is given to a Transformer, which already applies self-attention in the time dimension.

According to the conclusions we reached, we decided to build a block to adapt the speech representations to the Transformer, being as less intrusive as possible; creating what we named the Coupling network. The idea we had in mind was to use 1D convolutional layers to capture local patterns in the time axis and reduce the sequence length using stride. The S-Transformer decreases the time dimension by four, and we wanted to get the same reduction, so we planned to concatenate two 1D convolutional layers, with 512 channels at the input and the output (256 channels for PASE+), a kernel size of 3 and a stride of 2. The problem

[1] e.g. An utterance of 10 seconds is represented by a sequence of 1000 feature vectors.



(a) First approach (b) Second approach (c) Final approach

Figure A.0.1: Coupling network architectures.

we found is that the number of parameters needed was too big $(1,573,888)$, so we decided to substitute the second convolutional layer by an average pooling layer and to reduce the output channels to 256, recovering the 512 for the Transformer input with a linear layer (Fig. A.0.1a). With this configuration, we reduced the number of parameters to $525,056$.

However, the training did not work out as expected, and, after analysing the situation, we realised that we might be entrusting too much information to a simple average operation (Fig. A.0.2). We decided to go back to the approach with two convolutional layers, since we found it a better option to give the responsibility to learnable parameters (Fig. A.0.3). To keep the number of parameters low, we reduced the number of channels by two after each layer, getting 128 channels at the output of the convolutions (Fig. A.0.1b) and obtaining $557,952$ parameters to train.

Again, we were not satisfied with the training, so we hypothesised that maybe the bottleneck we created could be ruining the speech representations. As said before, keeping the 512 channels with normal 1D convolutional layers was not possible due to the high number of parameters it needed. Hence, we substituted them by Time-Channel separable convolutions, a type of convolution that is concatenation of a depthwise convolution and a pointwise convolution (Fig. A.0.1c). The first one is a convolutional layer that keeps the input channels at the output by not summing them up, which avoids the need for using as many filters as desired output channels. The second one is a convolutional layer with a kernel size of 1 and whose objective is to perform operations on the channels dimension, without affecting the temporal axis. Although this architecture is able to keep the number of channels from the speech representation without any bottleneck, it has a similar number of parameters than the previous approaches $(529,408)$.

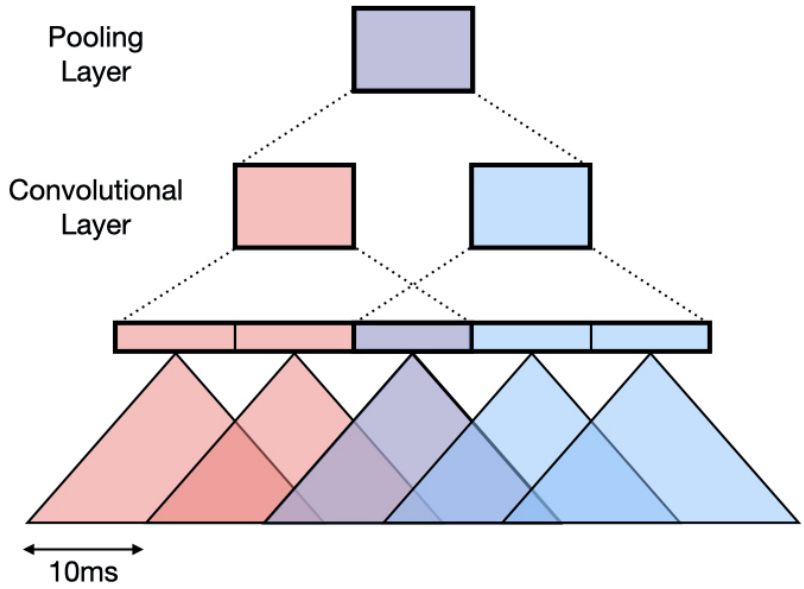


Figure A.0.2: Detail of the operations in the first Coupling network approach.

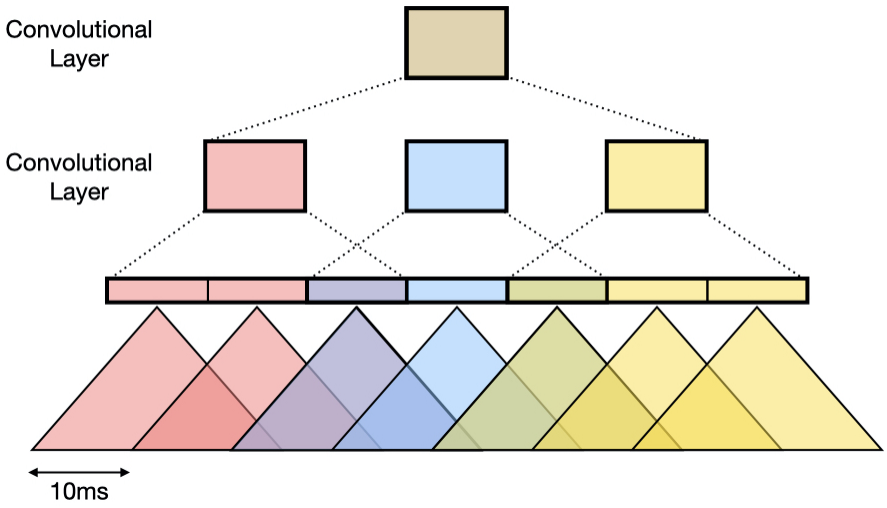


Figure A.0.3: Detail of the operations in the second and third Coupling network approaches.

Universitat Politècnica de Catalunya
Barcelona, 2020