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

THE UNIVERSITY of EDINBURGH
informatics

Towards Robust End-to-End Speech Translation

A Master's thesis submitted to the Faculty of the

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

by

Guillem Cortès Sebastià

In partial fulfillment of the requirements for the

Master's degree in Advanced Telecommunication Technologies

Supervisors:

Marta Ruiz Costa-Jussà, PhD. (UPC)
Barry Haddow, PhD. (UoE)

Barcelona, September 2020

"If a machine is expected to be infallible, it cannot also be intelligent."

Alan Turing

Abstract

Interest in speech-to-text translation systems has experienced a remarkable growth in recent years. The main motivation for this is the need to adapt to users the digital content they consume, for example, on social networks or video streaming platforms. In addition, nowadays we have high-quality automatic speech recognition and text translation systems which makes it the perfect time to investigate on speech translation systems. Traditionally cascade systems (ASR + MT) have worked best but great advances have recently been made in End-to-End systems which show their potential. This work is a study of the robustness of both systems, with the aim of being able to establish which approach is more resistant to noise. A series of experiments have been performed to determine which system is more robust. Both cascade and End-to-End systems have been trained with different noise levels using data from MuST-C En-Es, which contains 504 hours of speech, to study the difference in their performances. End-to-End systems have achieved a higher performance systematically. Despite of that, the behaviour of Cascade systems is pretty similar although they don't achieve the same performance. Moreover, training with noise provides a lot of stability and robustness.

Resum

L'interès pels sistemes de traducció de parla a text ha experimentat un creixement notable els darrers anys. La principal motivació que ha comportat aquest creixement és la necessitat d'adaptar a l'usuari el contingut digital que consumeix, per exemple, a les xarxes socials o a plataformes de vídeo *streaming*. A més, avui en dia tenim sistemes automàtics de reconeixement de parla i de traducció de text de gran qualitat la qual cosa fa que sigui el moment idoni per investigar sistemes de traducció de parla. Tradicionalment els sistemes en cascada (ASR+MT) són els que han funcionat millor però recentment s'han produït grans avenços en els sistemes *End-to-End*. Aquest treball és un estudi de la robustesa d'ambdós sistemes, amb l'objectiu de poder establir quina estratègia és més resistent a la presència de soroll. S'han realitzat una sèrie experiments entrenant sistemes en cascada i End-to-End, amb diferents nivells de soroll utilitzant les dades de MuST-C En-Es, que conté 504 hores de parla, per determinar quin sistema és més robust. Les conclusions que se'n poden extreure és que els sistemes End-to-End assoleixen un rendiment més elevat. Tot i això, el comportament davant el soroll és comparable als sistemes Cascada. Afegir que entrenar amb dades sorolloses aporta molta estabilitat i robustesa a qualsevol dels dos sistemes.

Resumen

El interés por los sistemas de traducción de habla a texto ha experimentado un crecimiento notable en los últimos años. La principal motivación que ha comportado este crecimiento es la necesidad de adaptar al usuario el contenido digital que consume, por ejemplo, en las redes sociales o plataformas de vídeo *streaming*. Además, hoy en día tenemos sistemas automáticos de reconocimiento de habla y de traducción de texto de gran calidad lo que hace que sea el momento idóneo para investigar sistemas de traducción de habla. Tradicionalmente los sistemas en cascada (ASR + MT) son los que han funcionado mejor pero recientemente se han producido grandes avances en los sistemas *End-to-End*. Este trabajo es un estudio de la robustez de ambos sistemas, con el objetivo de poder establecer qué estrategia es más resistente a la presencia de ruido. Se han realizado una serie de experimentos entrenando sistemas en cascada y End-to-End con diferentes niveles de ruido utilizando los datos de MuST-C En-Es, que contiene 504 horas de habla, para determinar qué sistema es más robusto. Los sistemas End-to-End consiguen un rendimiento más elevado y funcionan mejor. Sin embargo, el comportamiento delante señales ruidosas es muy parecido al de los sistemas en Cascada, aunque estos tienen un rendimiento pero. Añadir que entrenar con datos ruidosos aporta mucha estabilidad y robustez a cualquiera de los dos sistemas.

To Aliya, Nick and Talha.
Couldn't have asked for a better lockdown mates.

Acknowledgements

I would like to start thanking Barry Haddow for being an outstanding supervisor and always guiding me towards the correct direction. Thank you for being so responsive and available for a quick chat or question-answering session. Also, I would like to thank Marta Ruiz Costa-Jussà for contacting Barry and making this collaboration possible. Moltes gràcies Marta, tot el que he après i viscut aquests mesos m'acompanyarà la resta de la meva vida i és tot gràcies a tu.

To Gerard Gállego for the countless late-night sessions and for feeding me back with your motivation and our chats. Gràcies per totes les estones que hem passat junts, no hagués arribat aquí sense tu. To Carlos Escolano, gracias por estar siempre disponible para resolver cualquier duda y aportar/discutir ideas.

To Aliya, Nick, Talha and Momina for being responsible for many good memories of my stay in Edinburgh and keeping me entertained during lockdown. To Ramon for that road trip with all its conversations. Gràcies per obrir-te tant amb mi i deixar que jo també ho fes.

Last but not least, I would like to thank my family, for supporting me and giving me the opportunity to live in Edinburgh for almost 7 months; Jordi, Marta, m'heu fet el millor regal de la meva vida. Moltíssimes gràcies. I Mariona, gràcies per estar pendent de mi. And all my friends in Catalonia that encouraged me and supported me when the lockdown was not easy to handle; Albert, Joan, Jordi, gràcies pels podcasts i les trucades random; Bernat, Xavi, sou uns flipats i gràcies per això; Clara, gràcies per recordar-me que what you give is what you get i que tot torna. I Mar, gràcies per tot el viscut.

Table of contents

List of Figures

List of Tables

Nomenclature

ASR	Automatic Speech Recognition
BP	Brevity Penalty
BPE	Byte Pair Encoding
CTC	Connectionist Temporal Classification
E2E	End-to-End ST
KD	Knowledge Distillation
SLT	Spoken Language Translation
ST	Speech-to-text Translation
MT	Machine Translation
TTS	Text-To-Speech
WER	Word Error Rate

$\text{E2E}^{\diamondsuit}$	E2E with ASR pretraining and BPE
$\text{E2E} - \text{char}$	E2E trained at character-level
E2E_{raw}	E2E trained on raw data with BPE
$\text{E2E}_{low/mid/high}$	E2E trained on noisy data with BPE

Chapter 1

Introduction

Statement of purpose In an English-speaking world, we could think that translators are less necessary every day and that it is a dying profession. But the truth is completely opposite to this. With the increasing content generated around the world in social media, film industry, conferences, etc. Many of them decide to use English in order to reach the maximum possible audience but, the truth is that not everyone speaks English or has enough level to follow a speech. So automatic translation rules play an important role in spreading knowledge and information.

Traditionally, the best translation systems were compound by an Automatic Speech Recognition (ASR) and Machine Translation (MT) systems. Even though they still are very reliable systems, End-to-End approaches have become an interesting subject of research due to several benefits that present over the traditional cascade approaches:

- Can enable lower inference latency.
- It is easier to reduce the model size as it is only one integrated model.
- Avoid compounding errors from the ASR and MT models.
- Outperform cascade models when both are trained on Automatic Speech-to-text Translation (AST) parallel corpora.

However, cascade models are still outperforming end-to-end approaches due to the huge amount of data available compared to AST corpora accessible.

Use case In real-life scenarios, though, data is not always well recorded and there are a lot of artefacts that harm the Speech Translation system performance. Imagine for a second that we have developed an app for smartphones that translates speech to text for short sentences. Let's name this the tourist use-case. So the tourist wants to ask a local person where is the

closest bus stop. He opens the ST-app on his smartphone and starts talking, at the same time, a loud motorcycle passes next to him making a lot of noise.

Ambition So the question here is, does it make any difference to have an End-to-End ST or a cascade ST? Do they have a similar behaviour when they have to translate from a noisy signal? Are End-to-End systems more robust than cascade systems? Is the avoidance of compounding errors from ASR and MT models enough to make such a statement?

Thesis structure This dissertation is organised as follows: first I've done a superficial and quick introduction to which the problem is, and what are my plans to tackle it. Then, in Chapter §2 I introduce the transformer as well as ASR, MT and e2e systems very briefly. In Chapter §3 there's an exhaustive overview of which is the actual state-of-the-art and in which direction researchers on ST are working towards. Chapter §4 includes a list of all public ST datasets as well as list of tools I've used in this project and a breakdown of the architectures used. At *Experimental Framework* (Chapter §5) I explain all the experiments I've carried out, as well as data analysis or noise generation. After that, we only have to analyse the results of the experiments, draw conclusions and think about future lines of research (Chapters §6 and §7).

This work lies under the umbrella of ELITR — European Live Translator Project (ELITR, 2020) and it has been carried out at School of Informatics — University of Edinburgh, 2020.

Chapter 2

Background

In order to understand this work in its entirety we must first make sure that we are on the same page and that we think of the same thing when we talk about ASR, NMT or End-to-End. I also present the key points for understanding the architecture that supports every neural translation system today: the transformer. And what do we have to change if we want to use it with audio instead of text.

2.1 Systems overview

Because a picture is worth a thousand words, Figure 2.1.1 represents the project's playground where where I have beeen playing around trying to determine which system is more robust.

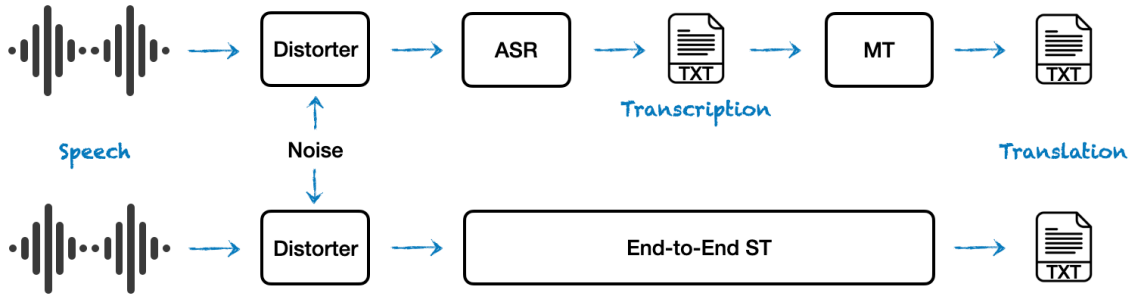


Figure 2.1.1: Cascade (top) and End-to-End (bottom) block diagrams

Cascade systems consist of different independent blocks, 2 at least, that we connect to get what we want. In Speech Translation we usually connect an Automatic Speech Recognition block with a Machine Translation one (nowadays the vast majority them are, in fact, Neural Machine Translation blocks). The ASR input can be raw speech directly, or a representation of it. It depends on each ASR. The ASR generates the transcriptions of the input signal in the same language. These transcriptions are used later as the input of the MT system, which translates the transcriptions to the target language.

Ent-to-End systems of one single block, so there are no intermediate steps. In ST, the E2E input is the raw audio — or again, it can be a representation of it — and the output is the translation of it. As it is noted in the Introduction §1, the belief is that not having an intermediate step makes the system more robust.

So, if we imagine each system as a black box, they are interchangeable because input and output are the same. This can be seen clearly in Figure 2.1.1. We have someone speaking producing speech, we record this speech but with it we are recording many other things like reflections, third-party noises, the envelope of the room/space where the speaker is (this is recorded indirectly) and many other things. So, to keep things simple, we can say that the original speech signal has been distorted by all this artefacts and that's actually the signal we will work with.

2.2 Transformer

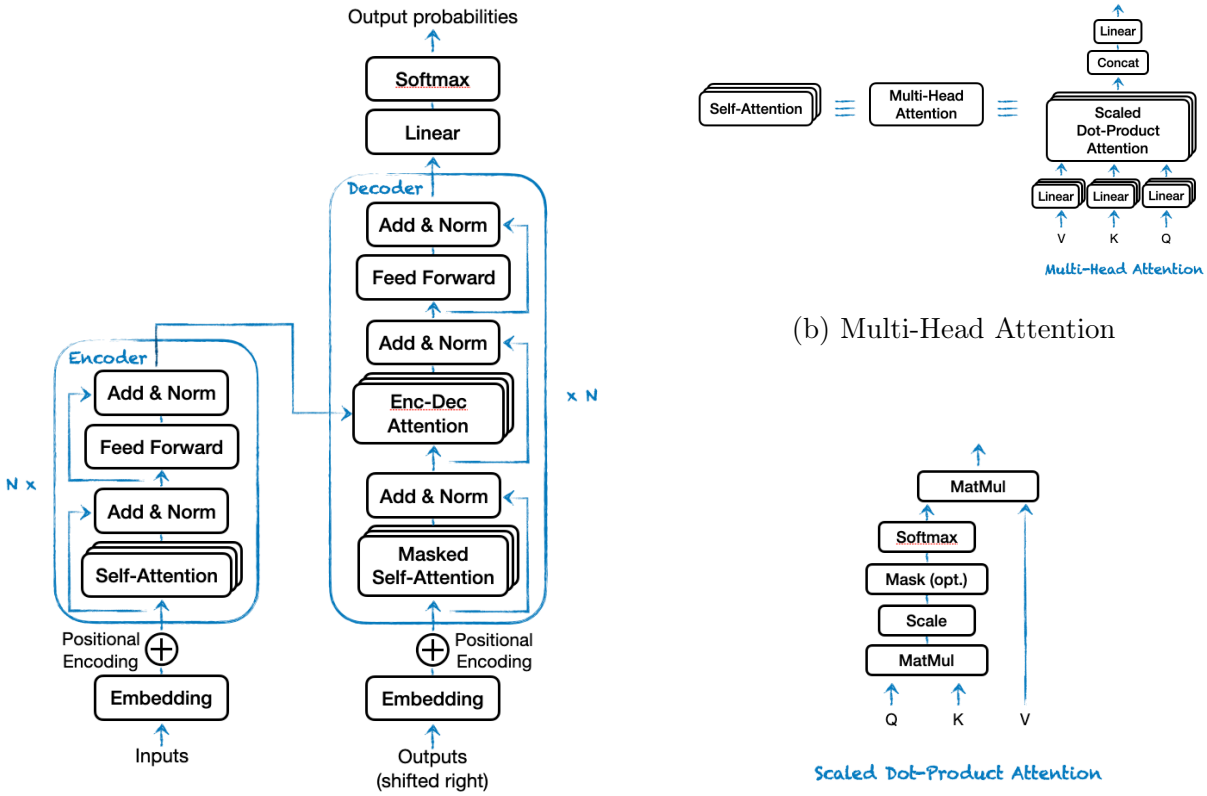
Transformer (Vaswani u. a., 2017) is a widely used sequence to sequence encoder-decoder architecture entirely based on attention networks which makes it really good in processing sequences. Attention is no other thing that a mechanism that looks at an input sequence and decides at each step which other parts of the sequence are relevant. Attention is weighing individual words in the input sequence according to the impact they make on the target sequence generation. The fact that attention weights are calculated using all the words in the input sequence at once facilitates parallelisation and that's one of the Transformers' strongest points. Figure 2.2.1 shows its full architecture.

So Transformer works perfectly when we want to process sequences of words, it was designed for it so that make sense. The problems start when we want to process speech with it. Then, the number of input tokens is much higher, causing a computational problem. Another issue that appears is that Transformer cannot model 2D dependencies over time and frequency in a spectrogram, so we have to introduce some elements in the network capable of doing that.

Last but not least, the absence of an explicit bias towards the local text — i.e. short-range dependencies between the input feature — is also something that requires an action.

The reason why this happens is that attention works really great for long-range dependencies, but it is not that good with short ones.

An architecture — *S-Transformer* — that solves all of this problems was proposed in *Adapting Transformer to End-to-end Spoken Language Translation* (Di Gangi u. a., 2019b) and it is explained in Section 4.1.



(a) Transformer architecture

(b) Multi-Head Attention

(c) Scaled Dot-Product Attention

Figure 2.2.1: Transformer architecture breakdown

Chapter 3

Literature Review

As I already said, the Speech Translation field has experienced growth in popularity in the last few years. The community has been focusing on achieving the best possible performance with the maximum data efficiency. To do so, there have been many different approaches, Knowledge Distillation, Data Augmentation, Pretraining, Multitask learning, Meta-learning, etc. But, even though there are a significant amount of research papers in this field, this thesis is the first work that focuses on the robustness of existing systems, to the best of my knowledge. This Chapter gathers the most remarkable works in Speech Translation and classifies them by approach.

Speech Translation and the End-to-EndPromise:Taking Stock of Where We Are (Sperber und Paulik, 2020) is an exceptional analysis of the actual (was published on April, 2020) situation of ST. They realised that to improve data efficiency, most end-to-end models employ techniques that re-introduce issues generally attributed to cascaded ST.

In the search for the best ST system, several new approaches, methods and techniques has been proposed. **Multi-task training and pretraining** were proposed as a way to incorporate additional ASRand MT data and reduce dependency on scarce end-to-end data (Weiss u. a., 2017) (Bérard u. a., 2018) (Bansal u. a., 2018). These approaches were not able to use ASR and MT data as loosley coupled cascade systems do, so other approaches emerged: **Data augmentation** (Pino u. a., 2019) by using Text-to-Speech from MT parallel corpus & MT translating the ASR transcript. **Knowledge Distillation** (Liu u. a., 2019) in which they aim that ST models work better if they have been trained with knowledge distillation from an MT system. So they developed a ST model that learned from ground truth translations and teacher model outputs. They obtained the highest BLEU score when learning only from the teacher. **Meta-learning** was proposed by (Indurthi u. a., 2020) and they claim to be state-of-the-art on ST. (Hsu u. a., 2020) they also use Meta-learning for low-resource ASR.

Other interesting works have been carried out by Fondazione Bruno Kessler — FBK. In *Adapting Transformer to End-to-end Spoken Language Translation* (Di Gangi u. a., 2019b) they present the S-Transformer, the system I have used as E2E and presented in Section 4.1. In *End-to-End Sp eech-Translation with Knowledge Distillation* (Gaido u. a., 2020) they focus on Transfer Learning (ASR pretraining and KD); Data Augmentation (SpecAugment, time stretch, synthetic data); Combining real and synthetic data in different domains; and Multitask learning using CTC loss.

Luckily, I could attend to ICASSP. Below I present the papers I found more interesting.

Analyzing ASR Pretraining for Low-Resource Speech-to-Text Translation (Stoian u. a., 2020). They found that best WER of the pre-trained ASR model is likely the best direct predictor of ST performance. They also claim that the most important thing in ASR pretraining is having a lot of data rather than having data from a close language. McCarthy et al. presented SKINAUGMENT (McCarthy u. a., 2020), which is state-of-the-art in data augmentation (better than SpecAugment (Park u. a., 2019)) They learn speaker representations with an autoencoder and create new audios by characterisin existing audios with a different speaker. *PASE+* (Ravanelli u. a., 2020). An improved version of PASE (Problem Agnostic Speech Encoder) for robust speech recognition in noisy and reverberant environments.

Chapter 4

Methodology

In order to determine which system is more robust to noise, Cascade systems or End-to-End, I need to know which systems (frameworks, architectures), dataset and toolkits I'm going to use. This chapter is a collection of available datasets, tools and architectures for Speech Translation. I also explain the reasons why I chose one tool or another one as well as the methodology that followed that decision.

4.1 S-Transformer

The main issue when we want to use a Transformer on speech data is that the Transformer's memory complexity produces a computational problem, because of Self-Attention's GPU complexity, which is quadratic in sequence length. This is usually tackled with downsampling methods and it can enable SLT training on GPUs. Another issue is that Transformer cannot model 2D dependencies over time and frequency in a spectrogram (Li u. a., 2016). To address this, 2D adaptation strategies of the Transformer encoder (Dong u. a., 2018) have been proposed and validated. The last big problem is the absence of an explicit bias towards the local text — i.e. short-range dependencies between the input feature. This has been addressed by explicitly modeling short-range dependencies for acoustic models either using hard masking (Povey u. a., 2018), or penalising the self-attention weighting based on the distance between input elements (Sperber u. a., 2018) so it penalizes the long-distance relations in favour of the closer ones.

S-Transformer (Di Gangi u. a., 2019b) is a variant of the Transformer that includes all these solutions to adapt the Transformer to SLT. Figure 4.1.1 shows the S-Transformer Encoder architecture where the first two CNNs capture local 2D-invariant features in the input, while the following two 2D self-attention layers model long-range context.

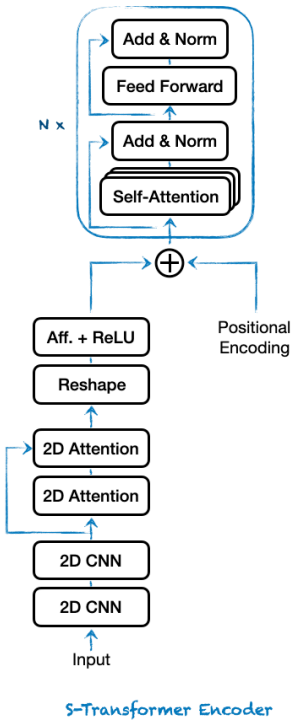


Figure 4.1.1: S-Transformer Encoder architecture

4.2 Toolkits

If you are planing to work in ST, the first thing you must know is that it is a multi-disciplinary field. So you will come across many speech processing, ASR toolkits on one hand and MT toolkits on the other hand. But the truth is that there aren't that many ST toolkits so here I summarise all toolkits I've used and my experience with them.

First, I wanted to use *ESPnet*[1] (Watanabe u. a., 2018) because it is a super complete toolkit with a lot of features and seems to be up-to-date and maintained. My experience is not as satisfactory as it seemed at first. The installation process is quite tedious and annoying, and I had some problems to set it up at university's servers. But after installing it succesfully, I couldn't replicate their baseline so I moved forward to *FBK-Fairseq-ST* (Di Gangi u. a., 2019b). It is an adaptation of *fairseq* (Ott u. a., 2019) for direct speech translation. The advantages of FBK-Fairseq-ST is that is based on fairseq, a consolidated sequence omdelling toolkit that runs on python and pytorch so that makes it really flexible and easy to tweak (because fairseq has been conceived to be very adaptable). The main drawback is that there isn't a community behind the project and might be difficult to solve issues that may arise. Luckily, all the issues were solved and I was able to train systems using FBK-Fairseq-ST tool.

[1] In addition to ESPnet, creators presented ESPnet-ST (Inaguma u. a., 2020) at ACL 2020 which has been designed for the quick development of speech-to-speech translation systems in a single framework.

For the cascade system I had to find a strong ASR with a trained model in English and an MT with En-Es trained model as well. So according to this, I have used *Quartznet* (Kriman u. a., 2020) which is an end-to-end neural acoustic model for ASR. QuartzNet's design is based on the Jasper (Li u. a., 2019) architecture, which is a convolutional model trained with Connectionist Temporal Classification (CTC) loss (Graves u. a., 2006) but in which they replaced the 1D convolutions with 1D time-channel separable convolutions, an implementation of depthwise separable convolutions. For the MT system I have used Bergamot translation models (statmt, 2020) for Marian (Junczys-Dowmunt u. a., 2018). These models have been trained on OPUS+OpenSubtitles+Paracrawl data, cleaned with rule-based and dual cross-entropy noise filtering, and trained with sampled back-translations (sBT).

Then, additional toolkits that I have used: moses (Koehn u. a., 2007) for tokenisation scripts, SoX (Bagwell, 2000) for adding noise to the dataset, xnmt (Neubig u. a., 2018) and librosa (McFee u. a., 2020) to generate de melspectograms and export them into *.h5* files.

To compute WER I used the asr-evaluation repository (Lambert, 2018) and for BLEU scores Sacrebleu (Post, 2018).

4.3 Datasets

Neural Networks need data, a lot of it, so it is important to use as much as it is available. And particularly in Speech-to-text Translation where there are less aligned and parallel corpora than there are for ASR or MT. For a corpus to be considered an ST dataset, it needs to include – at least – parallel data of audio in the source language and the written translations in the target language so it can be used to train an End-to-End system. Most of the times they also include the transcription in the original language, that it becomes really useful for ASR pretraining or training a cascade system. In this section, I describe the main ST datasets that are available up until today. I am presenting them in reverse chronological order, starting with the most recent ones. Later on in section 5.3, I will explain why I chose to **use *MuST-C En-ES* dataset**.

Publicly available SLT Datasets

CoVoST (Wang u. a., 2020a), created and released under CC0 license by Facebook AI, CoVoST (Common Voice Speech Translation) is a multilingual speech-to-text translation corpus from 11 languages to English. It has a lot of diversity due to the 11,000 speakers and more than 60 accents involved. Audios and transcriptions are from Common Voice dataset (Ardila u. a., 2019).

CoVoST 2 (Wang u. a., 2020b) expands on the CoVoST dataset conforming the largest multilingual ST dataset available to date. CoVoST 2 contains data that will facilitate translating 21 languages into English, as well as English into 15 languages.

EuroparlST (Iranzo-Sánchez u. a., 2020) corpus has been compiled using the debates held in the European Parliament between 2008 and 2012 resulting in a corpus with 6 European languages, for a total of 30 different translation directions. This corpus was created with the aim to be the reference dataset for SLT of languages other than English. Europarl-ST is the first fully self-contained, publicly available corpus with both, multiple (speech) source and target languages, which will also enable further research into multilingual SLT.

MuST-C (Di Gangi u. a., 2019a) is a multilingual corpus from English to 8 different languages. It was created pursuing high quality as well as large size, speaker variety (male/female, native/non-native) and coverage in terms of topics and languages. That's why they extracted the original audios from TED Talks (TED). They also provide a *yaml* file with the alignment between audio and text translations.

For all the experiments I have carried out in this project I have used the English to Spanish set from MuST-C. I chose MuST-C because, first, it is the dataset with the largest amount of SLT data English-Spanish, and I wanted to use either English, Spanish or Catalan due to they are the languages I can speak. Secondly, because its high speaker variety as well as all the different topics covered. And in addition, Mattia Di Gangi worked on its development and he released[2] as well a ST version of fairseq (Di Gangi u. a., 2019b).

Augmented Librispeech (Kocabiyikoglu u. a., 2018) is an extension of Librispeech (Panayotov u. a., 2015) dataset that gathers English audiobooks that translate into 1000 hours of speech. In Augmented Librispeech they collect French translations of that books and perform a bilingual text alignment from comparable chapters.

How2 (Sanabria u. a., 2018) is a large-scale dataset for multimodal understanding but, on top of that, it is multilingual as well. So because of that, we can use part of the available data and perfectly fits as an SLT dataset. All videos are in English and were extracted from *YouTube*, and the Translations are in Portuguese.

Fisher and CALLHOME (Post u. a., 2013) is actually a combination of two datasets: **Fisher** Spanish dataset consists of 819 transcribed conversations on an assortment of provided topics primarily between strangers. **CALLHOME** Spanish corpus comprises 120 transcripts of spontaneous conversations primarily between friends and family members.

(Paulik und Waibel, 2009) worked with a custom dataset that gathered 111 hours En-Es and 105 hours Es-En but, unfortunately, this data hasn't been published.

[2]S-Transformer github repository. `https://github.com/mattiadg/FBK-Fairseq-ST`

Chapter 5

Experimental Framework

Let's stop for a second and recapitulate. This project is a study of the robustness of the two more used approaches to tackle Speech-to-text Translation: Cascade and End-to-End. The motivation lies in the fact that one of the main reasons to use an End-to-End approach instead of the traditional ASR-MT is that the latter propagates the ASR errors and that harms the performance. End-to-End systems avoid compounding errors because they generate the translation text directly. But does this mean that End-to-End systems are more robust than the traditional cascade approach?

5.1 Experiments cookbook

In order to determine which experiments are necessary and which ones are the best we have to ask ourselves, How can we measure robustness? It is not an easy question to answer, for sure. So we can try to reformulate it: Which system is more robust to noise, cascade or End-to-End? Now, this question seems easier to answer. What we can do is to compare the performance of both approaches when we train or inference with noisy data. We can also establish different levels of noisiness and cross all possibilities. Table 5.1.1 summarises the notation I use to refer to each system configuration. Some considerations:

- E2E and ASR use the same S-transformer architecture, the only difference between them is the target data that in ASR is the transcriptions and in E2E is the translations.
- E2E$^{\diamond}$ uses the encoder of an ASR trained with the same configuration as pretraining.
- NeMo (Kuchaiev u. a., 2019) is Python toolkit developed by nvidia for creating AI applications. They provide a strong, ready to use ASR Quartznet Jasper models (Li u. a., 2019) trained on *Librispeech, Common Voice, Fisher, WSJ, Switchboard*.

- Bergamot (statmt, 2020) are Marian-NMT (Junczys-Dowmunt u. a., 2018) models trained on OPUS+OpenSubtitles+Paracrawl data, cleaned with rule-based and dual cross-entropy noise filtering, and trained with sampled back-translations (sBT).

System configuration	**Description**
$\text{E2E}_{\text{raw/low/mid/high}}$	End-to-End ST
$\text{E2E}^{\diamondsuit}_{\text{raw/low/mid/high}}$	End-to-End ST with ASR pretraining
ASR_{NeMo} + NMT	NeMo ASR + Bergamot NMT
$\text{ASR}_{\text{raw/low/mid/high}}$ + NMT	End-to-End ASR + Bergamot NMT

Table 5.1.1: Systems descriptions for experiments
$^{\diamondsuit}$ ASR pretraining

So basically, my experiments consist in training all these systems and generating translations by inferring *raw*, $Noisy_{low}$, $Noisy_{mid}$ and $Noisy_{high}$ data. Check section 5.3.2 for more details about the added noise and Chapter 6.

5.2 Data

In order to study the robustness of both approaches, it is crucial to work with languages I can speak – i.e. Catalan, Spanish and English – so I can evaluate the outputs manually to look for differences. Another important factor to pay special attention is the quality of the recordings and if this quality is consistent throughout all the dataset – since some experiments consist in adding noise to the original audio. Last but not least, it is also important the topics coverage and domain variety.

So, as I detailed above, it is crucial for me to work with languages I can speak, write and understand, so this restricts the data that I can use. In Figure 5.2.1 there's a detailed comparison between all available datasets with parallel translated data in any of the combinations between English, Spanish and Catalan. The two datasets with more data are *MuST-C En-Es* and *CoVoST-2 En-Ca*. I could have worked with both of them, but when I started this project on January 2020, *CoVoST-2* had not yet been published so I opted for MuST-C.

Data Analysis An important step that most of the times is overlooked in research is to analyse data you'll be working on. And this process is even more important when working with text. Every day we, humans, create and consume tons of text, so there's plenty of them available on internet and sometimes, very easy to get by scraping the web, and that's great news. But most of the times we assume that the data we are using is correct or has a good quality, and luckily most of the times will be like that, but there are others when the

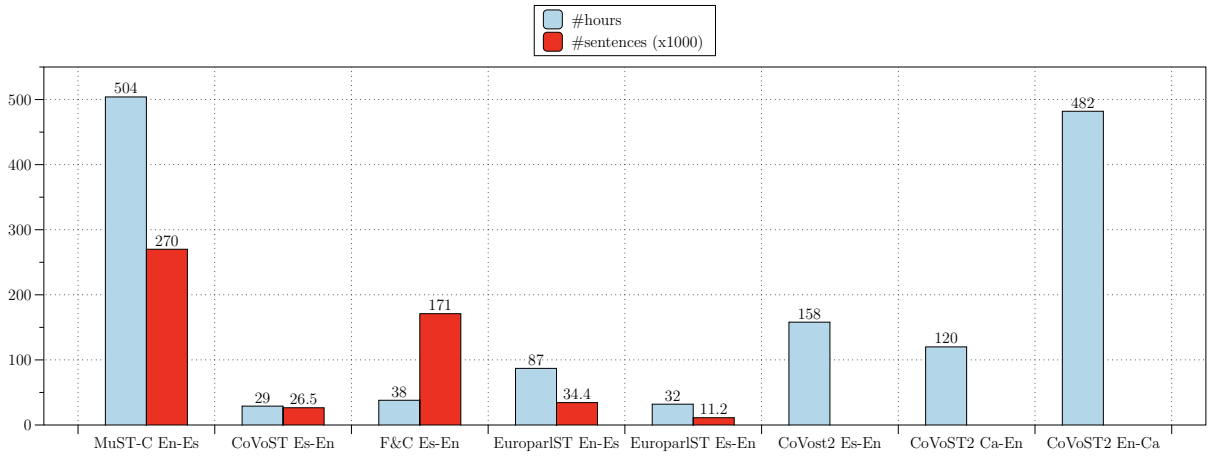


Figure 5.2.1: Available data from datasets in English, Spanish or Catalan

data won't be good. Per example, recently someone realised that many articles in Scots at Wikipedia[1] (McDonald, 2020) were written by an American Teenager who doesn't speak any Scots. Therefore, data analysis is always not only a good idea, but a necessary step.

Looking into MuST-C data, and En-Es subset more specifically, we see that, as well as pretty much everything in this life, pros and cons. The **advantages** of this dataset is the gender variability of speakers and their English level, accent. It also covers a lot of topics so it has a wide domain. The quality of the recordings is great as well. It is also one of the largest ST corpora publicly available. **On the other hand**, due to audios were extracted from TED Talks and each talk was in a different place, the quality of the recordings varies from one to another. There's also some voice overlapping when the presenter interacts with someone in the audience or there's an interview — this can be seen as an advantage, actually — and sometimes the audio comes from a video or audio played on stage at the talk.

Table 5.2.1 contains 6 sentences from *tst-COMMON.en*, which is the transcriptions test subset. We can appreciate that they contain some annotations about who said what in a theatrical script style, and of course, it is text that it's not being said. Another thing to take into consideration is that the alignments speech-text are not perfect. Sometimes due to audience applause but others are just simply wrong. Sometimes there's a lot of laughter or applause in a sentence, other times the speech is cut and misses the last word of the sentence. What also happens is that, just because it's a live, continued speech, the narrator stops in the middle of a sentence, repeat a word or the audience laughs overlapping the speech.

[1] `https://en.wikipedia.org/wiki/Main_Page`

# Sentence	Transcription
26	So get in the game. Save the shoes.
27	Thank you.
28	(Applause)
29	Bruno Giussani: Mark, Mark, come back.
30	Mark Bezos: Thank you.
31	I just came back from a community that holds the secret to human survival.

Table 5.2.1: Extract from *tst-COMMON.en* transcription

5.3 Data processing

5.3.1 Removing annotations

As we have seen in Table 5.2.1, the transcriptions, and therefore the translations, contain some annotations that might harm the performance of the ST system. To check this hypothesis, I carried out a simple experiment in which I removed the annotations from the dataset and trained an **E2E-char**$^{\diamond}_{raw}$ – End-to-End Speech-to-text Translation with ASR pretraining trained on raw data (meaning that is not noisy) and using character tokenisation — to compare its performance with the performance of the same system trained with original data. In Table 5.3.1 we can find the 5 most common annotations of *train* transcriptions that, just for reference, *train* subset has 5,181,350 words in 265,625 sentences. BLEU scores of the experiment can be found in Table 6.0.1.

Annotation	(Laughter)	(Applause)	CA:	(Music)	(Video)	BG:
Appearances	9,970	4,596	842	387	298	163

Table 5.3.1: Most common annotations on train.en transcriptions.

These annotations were retrieved using regular expressions, looking for words between parentheses, words of two characters preceding a colon, and some others that were manually inserted like *Woman, Man, Interviewer, Narrator*, etc. Annotations were removed from translations as well of all subsets. I named the resulting dataset *cleaned* and trained the **E2E-char**$^{\diamond}_{raw}$ with both original and cleaned data. Results in Table 6.0.1 show that these annotations harm the performance of the system and we can improve it by removing them.

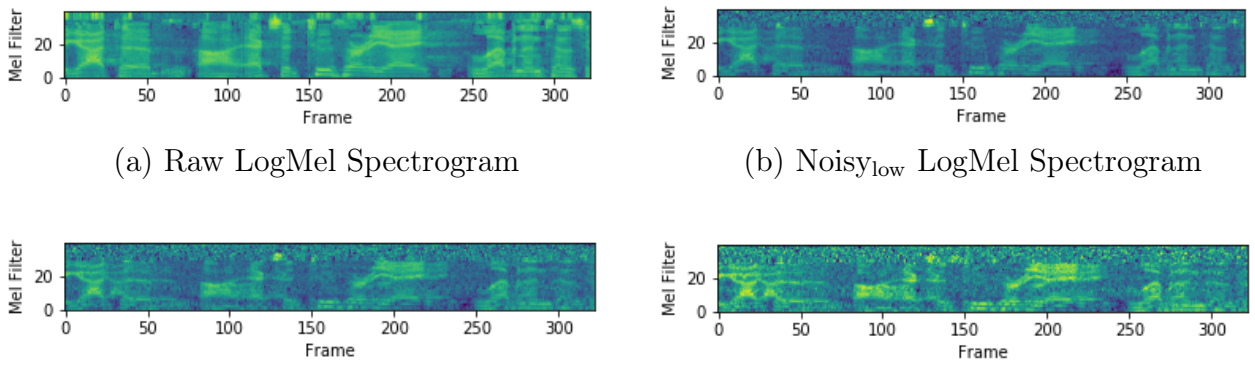
5.3.2 Adding noise

In Section 5.1 I have established that in order to evaluate the robustness of both cascade and E2E systems, we have to compare their performances when using noisy data. Here I detail which artefacts I applied to the dataset and why. In Section 5.4 the exact parameters and commands are detailed. In Appendix A, there are the exact code snippets that I've used.

When thinking about noise, inevitably, the first thing that comes to my mind is white noise. But here I refer to noisy data as data with some artefacts or distortions that reduce the intelligibility. Thus, we find that the most common distortions in speech recordings are reverberation, echo and clipping. With SoX[2] it is easy and straight forward to filter an audio using the effects *echo* and *reverb* (see Section 5.4 for the exact configuration). In order to add more complexity to the distortions I also added white noise. All of this raises the sequence volume so some clipping also appears although it's not that many, sequences that had most samples clipped where around 2,000 samples of a total of 160,000 (10 seconds sequence at 16kHz sampling rate). The difficulty about applying clipping manually is that each audio sequence has different gain levels, and that implies that is difficult to set a constant percentage of samples clipped throughout the whole dataset.

I wanted to generate three levels of noise: *low, mid* and *high*. The criteria I followed was that the noisiness levels of $\text{Noisy}_{\text{low}}$ have to be enough to notice the distortions when we listen to it, but we still can fully understand and comprehend everything that's being said. On the other hand $\text{Noisy}_{\text{high}}$ levels have to be high enough to reduce the intelligibility notoriously. $\text{Noisy}_{\text{mid}}$ levels were set right in between *low* and *high* levels. Figure 5.3.1 shows the spectrograms of the eleventh sentence of the train subset corresponding to "*We got to Exit 238, Lebanon, Tennessee.*"

[2]http://sox.sourceforge.net/sox.html



(a) Raw LogMel Spectrogram

(b) $\text{Noisy}_{\text{low}}$ LogMel Spectrogram

(c) $\text{Noisy}_{\text{mid}}$ LogMel Spectrogram

(d) $\text{Noisy}_{\text{high}}$ LogMel Spectrogram

Figure 5.3.1: Spectrograms corresponding to the eleventh sentence of train subset: *We got to Exit 238, Lebanon, Tennessee.*

5.4 Parameters and configuration

In this Section you can find all parameters and configurations that I've used in order to ease the reproducibility. In Appendix A, you can also found the exact code snippets that I've used.

Tokenisation

Before explaining configurations for noise generation, training and translations generation, I want to note that all E2E experiments rather than E2E-char have been trained with BPE (Sennrich u. a., 2015) tokenised data created with Sentencepiece (Kudo und Richardson, 2018). The data has been also tokenised and its punctuation normalised with moses (Koehn u. a., 2007). In addition, all ASR used character tokenisation following the guidelines of Mattia di Gangi, that includes removing the punctuation and lowercasing as well.

Noise Generation

The first distortion to apply is echoing. Echoes are reflected sound and can occur naturally amongst mountains and sometimes large buildings like churches. Essentially, after echoing a signal we have as a result the combination of two signals: the original one and the reflected. So with SoX we can set 4 parameters (Gain of input and output signals — these two can be set with almost every SoX effect — delay and decay. The time difference between the original signal and the reflection is the 'delay' (time), and the loudness of the reflected signal is the 'decay'. For all levels of noise I set a **gain-in** of 1 and **gain-out** of 0.8 and a **delay** of 150ms. So the only parameter that changes between $\text{Noisy}_{\text{low}}$, $\text{Noisy}_{\text{mid}}$ and $\text{Noisy}_{\text{high}}$ is the **decay** which I set levels of 0.02, 0.04 and 0.06 respectively.

After applying echo, I applied reverb which tweaks the persistence of sound after the sound is produced. Here the parameters we can set are reverberance, HF-damping, room-scale, stereo-depth, pre-delay and wet-gain. I only changed the **wet-gain** parameter, from 2 - 4 - 6 for $\text{Noisy}_{\text{low}}$ - $\text{Noisy}_{\text{mid}}$ - $\text{Noisy}_{\text{high}}$ configurations. The other parameters were left with the default value: **reverberance** 0.5 (50%), **HF-damping** 0.5 (50%), **room-scale** 1 (100%), **stereo-depth** 1 (100%), pre-delay 0ms.

Lastly, I added whitenoise, a random signal with constant power spectral density in order to add noise at all frequencies. Here, the only parameter to set is the white noise volume, which are 0.02, 0.04 and 0.06 for $\text{Noisy}_{\text{low}}$ - $\text{Noisy}_{\text{mid}}$ - $\text{Noisy}_{\text{high}}$ configurations.

Training configuration

For training the ASR or E2E systems, I follow the recommendations of Mattia di Gangi in (Di Gangi u. a., 2019b) of the optimal configuration he found for training the S-Transformer (big) that it is detailed in Chapter 4. It can also be checked in his post on Medium[3]. According to that, I kept the **batch size** to 512 by setting 8 **max-sentences** with an **update frequency** of 16 on 4 GPUs with 50 epochs limit. The clip threshold of gradients is set at 20 and adam optimizer with an inverse_sqrt learning rate scheduler that reduces the learning rate on plateau. Learning rate that starts with a warmup value of 3e-4 and 4000 updates and then is set at 5e-3. The flag *skip-invalid-size-inputs-valid-test* makes the system skip sentences that exceed one of these limits: 12000 tokens, 1400 source positions or 300 target positions. Finally, there's a 0.1 for both dropout rate and label smoothing, all evaluated with label_smoothed_cross_entropy criteria using a logarithmic distance penalty.

Translations Generation

The problem with the generate translations script is that if the model was trained with the *skip-invalid-size-inputs-valid-test* flag on, it fails to generate long sentences so we have two options here: First one is to also skip long sentences in generation. But this is something that's not desired since we want to generate an output for every single inferring sentence so we can evaluate them with the reference and compute metrics. The second option is to *overwrite* the **max-source-positions** and **max-target-positions** parameters setting them to a relatively high number - p.e. 100000 for source and 5000 for target.

[3]https://towardsdatascience.com/getting-started-with-end-to-end-speech-translation-3634c35a6561

Chapter 6

Results

In this chapter I present and discuss the results of the experiments done. The test subset used for all these experiments is MuST-C en-es tst-COMMON.es. BLEU scores are extracted using the tokenized and normalised version of the groundtruth — *target* as reference.

Annotations In Section 5.3.1 I explained that in order to determine the effect of the annotations on the E2E performance I trained two E2E systems with ASR pretraining on Raw data. The first one using original data as can be found in MuST-C and the second one removing the annotations. Table 6.0.1 shows that they affect and that we can get +0.4 BLEU by removing them.

E2E-char$^{\diamondsuit}_{raw}$	
Data	**BLEU**
Original	21.1
Cleaned	**21.5**

Table 6.0.1: BLEU (↑) of **E2E-char**$^{\diamondsuit}_{raw}$ trained with original and *cleaned* MuST-C data
$^{\diamondsuit}$ ASR pretraining

Char vs BPE Because traditionally ASR systems train and generate at character level, the first ST systems also did (Bérard u. a., 2018) but recent studies (Gaido u. a., 2020) showed that BPE can outperform them. In order to evaluate which tokenisation is better when training End-to-End systems, I have trained some models using both approaches to see which one performs better. I wanted to test them in noisy environments/conditions so I trained them with noisy data — this time only with white noise — to see if there is any difference between them. As you can see in Table 6.0.2, I've also used raw and noisy data in inference, and compared the performance with a cascade system too.

The conclusion is that BPE works better, but not that much (with the exception of E2E-char trained with noise). Keep in mind that these E2E$^{\diamond}$ systems are with ASR pretraining and E2E without it.

		E2E		**E2E$^{\diamond}$**		**ASR+NMT**
Train	**Inference**	**CHAR**	**BPE**	**CHAR**	**BPE**	**(Cascade)**
Raw	Raw	17.1	17.4	**21.1**	**21.1**	18.9
	Noise	8.9	7.5	**11.7**	10	10.1
Noise	Raw	0	12.5	18.4	**19.7**	17.1
	Noise	0	11.6	17.7	**18.8**	15.8

Table 6.0.2: BLEU (↑) char vs BPE tokenisation
$^{\diamond}$ ASR pretraining

BPE parameters optimisation The training parameters of the experiments of the table above are the same for both char and BPE approaches, but my intuition is that they shouldn't be the same. Following Gaido's training configuration in (Gaido u. a., 2020), I trained a model analogous to E2E$_{raw}$. Basically, the paramaters that differ from one to the other are the Adam optimer's betas (0.9, 0.98), the learing rate (5e-4) and warmup-updates (5000). Table 6.0.3 shows the BLEU scores on tst-COMMON subset. As you can appreciate, the performance is worse so I will keep using the default configuration detailed in Section 5.4.

System	**Raw**	**Noise**
E2E$_{raw}$	17.4	7.5
E2E-optim$_{raw}$	14.9	6.9

Table 6.0.3: BLEU (↑) for different training parameters when using BPE

Noise Robustness ASR Before studying the robustness of End-to-End systems, first I did some experiments on ASR. For that, I used the same S-Transformer architecture (see Figure 4.1.1) that I use to train End-to-End models or the ASR pretraining. I trained these ASR with the three levels of noisiness I established in Section 5.3.2.

Figure 6.0.1 helps to visualise and understand the performance consistency among all types of noise. I will like to point out that ASR$_{\text{high}}$ is the system more consistent, which makes sense because it is the approach that has been trained with more noise, but ASR$_{\text{mid}}$ achieves an impressive performance that is actually better than ASR$_{\text{low}}$ for *low* noise levels, and almost the same for *raw* data. Check Table 6.0.4 for the exact BLEU scores.

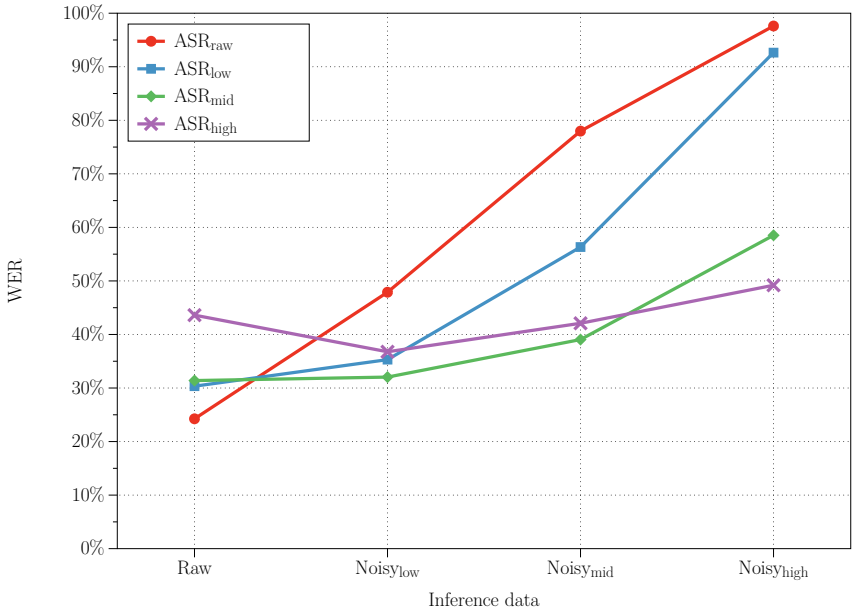


Figure 6.0.1: ASR Word Error Rate (↓)

	Raw	**Noisy**$_{low}$	**Noisy**$_{mid}$	**Noisy**$_{high}$
ASR$_{raw}$	24.25	47.89 (+23.64)	77.98 (+53.73)	97.61 (+73.36)
ASR$_{low}$	30.34	35.30 (+4.96)	56.33 (+25.99)	92.64 (+62.3)
ASR$_{mid}$	31.39	32.04 (+0.65)	39.05 (+7.66)	58.52 (+27.13)
ASR$_{high}$	43.61	36.77 (-6.84)	42.09 (-1.52)	49.19 (-5.58)

Table 6.0.4: WER (%, ↓) of different ASR configurations

End-to-End vs Cascade Once we have seen how ASR behaviour to the presence of noise, let's see how ST systems do. For that, I trained several models with different configurations and generate from different levels of noise as well. Figure 6.0.2 shows the performance curve of each system when we infer from different levels of noisiness. The exact BLEU scores can be found in Table 6.0.5. As a reference, MuST-C baseline on En-Es data is 18.20 BLEU (Di Gangi u. a., 2019a). Di Gangi et al., in their work *Adapting Transformer to End-to-end Spoken Language Translation* (Di Gangi u. a., 2019b) note a BLEU score of 22.5 performed by a cascade system.

What these experiments show is that all E2E$^{\diamondsuit}$ (End-to-End ST with ASR pretraining) systems outperform the cascade equivalent system, even though if that cascade uses an ASR trained with an S-transformer on MuST-C En-Es. The most probable thing happening here is that first, the domain is really important — even more than the amount of data — so it is crucial to train with data of the same domain you will be testing later. And then, the ASR might not be as strong as they can, so it's not a really good ASR in terms of performance.

If we now look at performance consistency, it happens the same exact thing as happened with ASR. $E2E^{\diamondsuit}_{high}$ is the most consistent system but $E2E^{\diamondsuit}_{mid}$ seems to achieve a really nice trade-off between consistency and performance, outperforming $E2E^{\diamondsuit}_{low}$. It is also worth mentioning that for each configuration, both the E2E and Cascade system curves are parallel, so it is fair to say that they behave really similarly in the presence of noise.

Last but not least, ASR_{NeMo} + NMT, the only system that hasn't seen any MuST-C data, performs pretty well and better than the ASR_{raw} model wihout seeing any data from TED Talks.

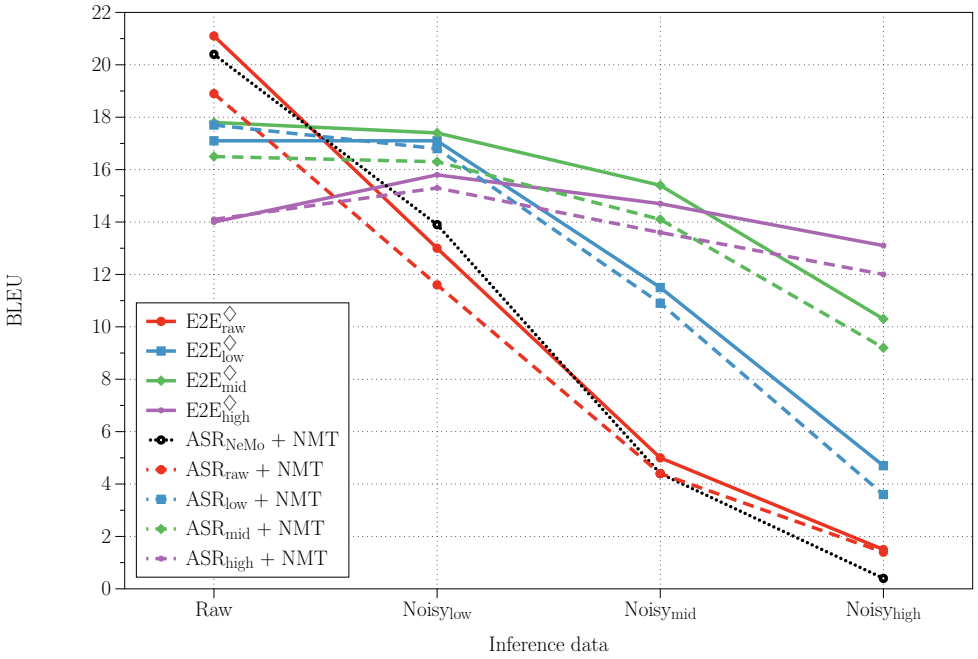


Figure 6.0.2: BLEU (↑) for all systems comparison
◇ ASR pretraining

System configuration	Raw	$Noisy_{low}$	$Noisy_{mid}$	$Noisy_{high}$
$E2E^{\diamondsuit}_{raw}$	**21.1**	13	5	1.5
$E2E^{\diamondsuit}_{low}$	17.1	17.1	11.5	4.7
$E2E^{\diamondsuit}_{mid}$	17.8	**17.4**	**15.4**	10.3
$E2E^{\diamondsuit}_{high}$	14	15.8	14.7	**13.1**
ASR_{NeMo} + NMT	**20.4**	13.9	4.4	0.4
ASR_{raw} + NMT	18.9	11.6	4.4	1.4
ASR_{low} + NMT	17.7	**16.8**	10.9	3.6
ASR_{mid} + NMT	16.5	16.3	**14.1**	9.2
ASR_{high} + NMT	14.1	15.3	13.6	**12**

Table 6.0.5: BLEU (↑) for all systems comparison
◇ ASR pretraining

Chapter 7

Conclusions and future work

The scope of this work was to shed light into the robustness of speech translation systems. More specifically, between End-to-End systems and cascade. As it is stated in the Introduction §1, the motivation appeared as a consequence of what seems to be accepted in the scientific community that End-to-End systems are more robust because among other things, avoid compounding errors from the ASR and MT models. The truth is that it seems a valid hypothesis, but have we — the scientific community — verified it's veracity? As it has been detailed in the Literature Review §3, there's not much work done in the field, and this hypothesis still has to be validated.

In my attempt to find which of the two systems is most robust, I have used several tools (§4) and read a lot about the state of art in Speech Translation. I've also been able to attend to ICASSP, one of the most important conferences in Speech, Acoustics and Signal Processing. All of this has made me learn a lot about Speech Processing and also MT, which the latter was quite unknown to me, and has given me enough knowledge to design experiments to validate the original hypothesis (§5). In this section I present the conclusions I have come up with alongside posible further work/research.

End-to-End systems perform better than cascade systems (§6) if the latter haven't been trained with similar data to the one we will evaluate the system with. But both E2E and cascade systems have parallel curves so they have a similar behaviour in noisy conditions. So it is not true that E2E systems are more robust than cascade systems. They have other advantages which can make them better than cascade systems, but in terms of robustness, they are very comparable. For the future, would be great to add sporadic (non-constant) noise, and see if there's any differences between approaches.

Training with noisy data gives consistency to system performance, and it is better to train with more noise than expected, not too much, though. $\text{E2E}^{\diamondsuit}_{\text{mid}}$ performing better than $\text{E2E}^{\diamondsuit}_{\text{low}}$ (and the same for their analogous ASRs) is a pleasant surprise that demonstrates that training with a significant level of noise does not harm the performance in raw conditions too much but increases it a lot in noisy conditions.

Having a lot of data that you can use for training is nice and it is always a good idea to use as much as you can, but training with data from the same domain is also super important. And even more important if the dataset has a different nature than the data used for training. By nature I refer to the fact that MuST-C used speeches from TED Talks, that involves some things: audience, interviews, speeches overlapping. It also means that it is a rehearsed speech, but can fluctuate a lot in intonation, prosody, etc. An interesting experiment for the future is to train a strong ASR and strong MT but also including MuST-C data in their training, to actually see the difference in performance with E2E system trained only on MuST-C.

Bibliography

[Ardila u. a. 2019] Ardila, Rosana ; Branson, Megan ; Davis, Kelly ; Henretty, Michael ; Kohler, Michael ; Meyer, Josh ; Morais, Reuben ; Saunders, Lindsay ; Tyers, Francis M. ; Weber, Gregor: Common voice: A massively-multilingual speech corpus. In: *arXiv preprint arXiv:1912.06670* (2019)

[Bagwell 2000] Bagwell, Chris: *SoX - Sound eXchange*. `http://sox.sourceforge.net/`. 2000. – [Online; accessed 30-August-2020]

[Bansal u. a. 2018] Bansal, Sameer ; Kamper, Herman ; Livescu, Karen ; Lopez, Adam ; Goldwater, Sharon: Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In: *arXiv preprint arXiv:1809.01431* (2018)

[Bérard u. a. 2018] Bérard, Alexandre ; Besacier, Laurent ; Kocabiyikoglu, Ali C. ; Pietquin, Olivier: End-to-end automatic speech translation of audiobooks. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2018, S. 6224–6228

[Di Gangi u. a. 2019a] Di Gangi, Mattia A. ; Cattoni, Roldano ; Bentivogli, Luisa ; Negri, Matteo ; Turchi, Marco: MuST-C: a multilingual speech translation corpus. In: *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* Association for Computational Linguistics (Veranst.), 2019, S. 2012–2017

[Di Gangi u. a. 2019b] Di Gangi, Mattia A. ; Negri, Matteo ; Turchi, Marco: Adapting Transformer to end-to-end spoken language translation. In: *INTERSPEECH 2019* International Speech Communication Association (ISCA) (Veranst.), 2019, S. 1133–1137

[Dong u. a. 2018] Dong, Linhao ; Xu, Shuang ; Xu, Bo: Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2018, S. 5884–5888

[ELITR 2020] ELITR: *European Live Translator*. `https://elitr.eu/`. 2020. – [Online; accessed 31-August-2020]

[Gaido u. a. 2020] Gaido, Marco ; Di Gangi, Mattia A. ; Negri, Matteo ; Turchi, Marco: End-to-End Speech-Translation with Knowledge Distillation: FBK@ IWSLT2020. In: *arXiv preprint arXiv:2006.02965* (2020)

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

[Hsu u. a. 2020] Hsu, Jui-Yang ; Chen, Yuan-Jui ; Lee, Hung-yi: Meta learning for end-to-end low-resource speech recognition. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 7844–7848

[Inaguma u. a. 2020] Inaguma, Hirofumi ; Kiyono, Shun ; Duh, Kevin ; Karita, Shigeki ; Yalta, Nelson ; Hayashi, Tomoki ; Watanabe, Shinji: ESPnet-ST: All-in-One Speech Translation Toolkit. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online : Association for Computational Linguistics, Juli 2020, S. 302–311. – URL `https://www.aclweb.org/anthology/2020.acl-demos.34`

[Indurthi u. a. 2020] Indurthi, Sathish ; Han, Houjeung ; Lakumarapu, Nikhil K. ; Lee, Beomseok ; Chung, Insoo ; Kim, Sangha ; Kim, Chanwoo: End-end Speech-to-Text Translation with Modality Agnostic Meta-Learning. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 7904–7908

[Iranzo-Sánchez u. a. 2020] Iranzo-Sánchez, Javier ; Silvestre-Cerdà, Joan A. ; Jorge, Javier ; Roselló, Nahuel ; Giménez, Adrià ; Sanchis, Albert ; Civera, Jorge ; Juan, Alfons: Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 8229–8233

[Junczys-Dowmunt u. a. 2018] Junczys-Dowmunt, Marcin ; Grundkiewicz, Roman ; Dwojak, Tomasz ; Hoang, Hieu ; Heafield, Kenneth ; Neckermann, Tom ; Seide, Frank ; Germann, Ulrich ; Fikri Aji, Alham ; Bogoychev, Nikolay ; Martins, André F. T. ; Birch, Alexandra: Marian: Fast Neural Machine Translation in C++. In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia : Association for Computational Linguistics, July 2018, S. 116–121. – URL `http://www.aclweb.org/anthology/P18-4020`

[Kocabiyikoglu u. a. 2018] Kocabiyikoglu, Ali C. ; Besacier, Laurent ; Kraif, Olivier:

Augmenting librispeech with french translations: A multimodal corpus for direct speech translation evaluation. In: *arXiv preprint arXiv:1802.03142* (2018)

[Koehn u. a. 2007] Koehn, Philipp ; Hoang, Hieu ; Birch, Alexandra ; Callison-Burch, Chris ; Federico, Marcello ; Bertoldi, Nicola ; Cowan, Brooke ; Shen, Wade ; Moran, Christine ; Zens, Richard u. a.: Moses: Open source toolkit for statistical machine translation. In: *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions* Association for Computational Linguistics (Veranst.), 2007, S. 177–180

[Kriman u. a. 2020] Kriman, Samuel ; Beliaev, Stanislav ; Ginsburg, Boris ; Huang, Jocelyn ; Kuchaiev, Oleksii ; Lavrukhin, Vitaly ; Leary, Ryan ; Li, Jason ; Zhang, Yang: Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 6124–6128

[Kuchaiev u. a. 2019] Kuchaiev, Oleksii ; Li, Jason ; Nguyen, Huyen ; Hrinchuk, Oleksii ; Leary, Ryan ; Ginsburg, Boris ; Kriman, Samuel ; Beliaev, Stanislav ; Lavrukhin, Vitaly ; Cook, Jack ; Castonguay, Patrice ; Popova, Mariya ; Huang, Jocelyn ; Cohen, Jonathan M.: *NeMo: a toolkit for building AI applications using Neural Modules*. 2019

[Kudo und Richardson 2018] Kudo, Taku ; Richardson, John: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: *arXiv preprint arXiv:1808.06226* (2018)

[Lambert 2018] Lambert, Ben: *ASR-evaluation: Python module for evaluting ASR hypotheses (i.e. word error rate and word recognition rate).* `https://github.com/belambert/asr-evaluation`. 2018. – [Online; accessed 30-August-2020]

[Li u. a. 2019] Li, Jason ; Lavrukhin, Vitaly ; Ginsburg, Boris ; Leary, Ryan ; Kuchaiev, Oleksii ; Cohen, Jonathan M. ; Nguyen, Huyen ; Gadde, Ravi T.: Jasper: An end-to-end convolutional neural acoustic model. In: *arXiv preprint arXiv:1904.03288* (2019)

[Li u. a. 2016] Li, Jinyu ; Mohamed, Abdelrahman ; Zweig, Geoffrey ; Gong, Yifan: Exploring multidimensional LSTMs for large vocabulary ASR. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2016, S. 4940–4944

[Liu u. a. 2019] Liu, Yuchen ; Xiong, Hao ; He, Zhongjun ; Zhang, Jiajun ; Wu, Hua ; Wang, Haifeng ; Zong, Chengqing: End-to-end speech translation with knowledge distillation. In: *arXiv preprint arXiv:1904.08075* (2019)

[McCarthy u. a. 2020] McCarthy, Arya D. ; Puzon, Liezl ; Pino, Juan: SkinAugment: auto-encoding speaker conversions for automatic speech translation. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 7924–7928

[McDonald 2020] McDonald, Karl: *Scots Wikipedia taken over by American teenager who wrote thousands of 'very odd' articles without learning language.* https://inews.co.uk/news/scotland/scots-wikipedia-language-articles-native-speaker-mistakes-610689. 2020. – [Online; accessed 27-August-2020]

[McFee u. a. 2020] McFee, Brian ; Lostanlen, Vincent ; Metsai, Alexandros ; McVicar, Matt ; Balke, Stefan ; Thomé, Carl ; Raffel, Colin ; Zalkow, Frank ; Malek, Ayoub ; Dana ; Lee, Kyungyun ; Nieto, Oriol ; Mason, Jack ; Ellis, Dan ; Battenberg, Eric ; Seyfarth, Scott ; Yamamoto, Ryuichi ; Choi, Keunwoo ; viktorandreevichmorozov ; Moore, Josh ; Bittner, Rachel ; Hidaka, Shunsuke ; Wei, Ziyao ; nullmightybofo ; Hereñú, Darío ; Stöter, Fabian-Robert ; Friesch, Pius ; Weiss, Adam ; Vollrath, Matt ; Kim, Taewoon: *librosa/librosa: 0.8.0.* Juli 2020. – URL https://doi.org/10.5281/zenodo.3955228

[Neubig u. a. 2018] Neubig, Graham ; Sperber, Matthias ; Wang, Xinyi ; Felix, Matthieu ; Matthews, Austin ; Padmanabhan, Sarguna ; Qi, Ye ; Sachan, Devendra S. ; Arthur, Philip ; Godard, Pierre ; Hewitt, John ; Riad, Rachid ; Wang, Liming: XNMT: The eXtensible Neural Machine Translation Toolkit. In: *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase.* Boston, March 2018

[Ott u. a. 2019] Ott, Myle ; Edunov, Sergey ; Baevski, Alexei ; Fan, Angela ; Gross, Sam ; Ng, Nathan ; Grangier, David ; Auli, Michael: fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In: *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019

[Panayotov u. a. 2015] Panayotov, Vassil ; Chen, Guoguo ; Povey, Daniel ; Khudanpur, Sanjeev: Librispeech: an asr corpus based on public domain audio books. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2015, S. 5206–5210

[Park u. a. 2019] Park, Daniel S. ; Chan, William ; Zhang, Yu ; Chiu, Chung-Cheng ; Zoph, Barret ; Cubuk, Ekin D. ; Le, Quoc V.: Specaugment: A simple data augmentation method for automatic speech recognition. In: *arXiv preprint arXiv:1904.08779* (2019)

[Paulik und Waibel 2009] Paulik, Matthias ; Waibel, Alex: Automatic translation from

parallel speech: Simultaneous interpretation as mt training data. In: *2009 IEEE Workshop on Automatic Speech Recognition & Understanding* IEEE (Veranst.), 2009, S. 496–501

[Pino u. a. 2019] Pino, Juan ; Puzon, Liezl ; Gu, Jiatao ; Ma, Xutai ; McCarthy, Arya D. ; Gopinath, Deepak: Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In: *arXiv* (2019), S. arXiv–1909

[Post 2018] Post, Matt: A Call for Clarity in Reporting BLEU Scores. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels : Association for Computational Linguistics, Oktober 2018, S. 186–191. – URL `https://www.aclweb.org/anthology/W18-6319`

[Post u. a. 2013] Post, Matt ; Kumar, Gaurav ; Lopez, Adam ; Karakos, Damianos ; Callison-Burch, Chris ; Khudanpur, Sanjeev: Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus. In: *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*. Heidelberg, Germany, December 2013

[Povey u. a. 2018] Povey, Daniel ; Hadian, Hossein ; Ghahremani, Pegah ; Li, Ke ; Khudanpur, Sanjeev: A time-restricted self-attention layer for asr. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2018, S. 5874–5878

[Ravanelli u. a. 2020] Ravanelli, Mirco ; Zhong, Jianyuan ; Pascual, Santiago ; Swietojanski, Pawel ; Monteiro, Joao ; Trmal, Jan ; Bengio, Yoshua: Multi-task self-supervised learning for Robust Speech Recognition. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 6989–6993

[Sanabria u. a. 2018] Sanabria, Ramon ; Caglayan, Ozan ; Palaskar, Shruti ; Elliott, Desmond ; Barrault, Loïc ; Specia, Lucia ; Metze, Florian: How2: a large-scale dataset for multimodal language understanding. In: *arXiv preprint arXiv:1811.00347* (2018)

[Sennrich u. a. 2015] Sennrich, Rico ; Haddow, Barry ; Birch, Alexandra: Neural machine translation of rare words with subword units. In: *arXiv preprint arXiv:1508.07909* (2015)

[Sperber u. a. 2018] Sperber, Matthias ; Niehues, Jan ; Neubig, Graham ; Stüker, Sebastian ; Waibel, Alex: Self-attentional acoustic models. In: *arXiv preprint arXiv:1803.09519* (2018)

[Sperber und Paulik 2020] Sperber, Matthias ; Paulik, Matthias: Speech Transla-

tion and the End-to-End Promise: Taking Stock of Where We Are. In: *arXiv preprint arXiv:2004.06358* (2020)

[statmt 2020] STATMT: *Bergamot translation models*. `http://statmt.org/bergamot/models/`. 2020. – [Online; accessed 30-August-2020]

[Stoian u. a. 2020] STOIAN, Mihaela C. ; BANSAL, Sameer ; GOLDWATER, Sharon: Analyzing ASR pretraining for low-resource speech-to-text translation. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE (Veranst.), 2020, S. 7909–7913

[TED] TED: *TED Talks*. `https://ted.com/talks`, note = [Online; accessed 25-August-2020]

[Vaswani u. a. 2017] VASWANI, Ashish ; SHAZEER, Noam ; PARMAR, Niki ; USZKOREIT, Jakob ; JONES, Llion ; GOMEZ, Aidan N. ; KAISER, Łukasz ; POLOSUKHIN, Illia: Attention is all you need. In: *Advances in neural information processing systems*, 2017, S. 5998–6008

[Wang u. a. 2020a] WANG, Changhan ; PINO, Juan ; WU, Anne ; GU, Jiatao: Covost: A diverse multilingual speech-to-text translation corpus. In: *arXiv preprint arXiv:2002.01320* (2020)

[Wang u. a. 2020b] WANG, Changhan ; WU, Anne ; PINO, Juan: CoVoST 2: A Massively Multilingual Speech-to-Text Translation Corpus. In: *arXiv preprint arXiv:2007.10310* (2020)

[Watanabe u. a. 2018] WATANABE, Shinji ; HORI, Takaaki ; KARITA, Shigeki ; HAYASHI, Tomoki ; NISHITOBA, Jiro ; UNNO, Yuya ; ENRIQUE YALTA SOPLIN, Nelson ; HEYMANN, Jahn ; WIESNER, Matthew ; CHEN, Nanxin ; RENDUCHINTALA, Adithya ; OCHIAI, Tsubasa: ESPnet: End-to-End Speech Processing Toolkit. In: *Proceedings of Interspeech*, URL `http://dx.doi.org/10.21437/Interspeech.2018-1456`, 2018, S. 2207–2211

[Weiss u. a. 2017] WEISS, Ron J. ; CHOROWSKI, Jan ; JAITLY, Navdeep ; WU, Yonghui ; CHEN, Zhifeng: Sequence-to-sequence models can directly translate foreign speech. In: *arXiv preprint arXiv:1703.08581* (2017)

Appendix A

Code

BPE tokenisation

```
spm_train --input=$infile --model_prefix=$lang --model_type=bpe
spm_encode --model=$lang.model --output_format=piece < $infile >
    $outfile.bpe
spm_decode --model=$sentencepiece-files/$lang.model --input_format=
    piece < $infile.bpe > $outfile.word
```

Code A.0.1: BPE tokenisation with sentencepiece

Noise Generation

```
#Noisy Low
sox $file $tmpfile echo 1 0.8 150 0.2 reverb 0.5 0.5 1 1 0 2
sox $tmpfile -p synth whitenoise vol 0.02 | sox -m $tmpfile -
    $outfile
#Noisy Mid
sox $file $tmpfile echo 1 0.8 150 0.4 reverb 0.5 0.5 1 1 0 4
sox $tmpfile -p synth whitenoise vol 0.04 | sox -m $tmpfile -
    $outfile
#Noisy High
sox $file $tmpfile echo 1 0.8 150 0.6 reverb 0.5 0.5 1 1 0 6
sox $tmpfile -p synth whitenoise vol 0.06 | sox -m $tmpfile -
    $outfile
```

Code A.0.2: Add noise

```
#$FBKFairseqST path to https://github.com/mattiadg/FBK-Fairseq-ST.git

CUDA_VISIBLE_DEVICES=$GPUS python $FBKFairseqST/generate.py \
    $DATA/$LANG-bin/ \
    --path $models/checkpoint_best.pt \
    --audio-input --max-source-positions 100000 \
    --max-target-positions 5000 \
    --gen-subset $subset > $outfile
python $FBKFairseqST/scripts/sort-sentences.py $outfile 5 > $outfile.lines
#BPE - sentencepiece
spm_decode --model=$sentencepiece_files/es.model --input_format=piece
    < $outfile.lines > $outfile.lines.word
#CHAR - character tokenisation
bash $FBKFairseqST/scripts/extract_words.sh $outfile.lines
```

Code A.0.3: Generate translations

```
CUDA_VISIBLE_DEVICES=$GPUS python $FBKFairseqST/train.py $MUSTC/en-es/data-st/bpe/$LANG-bin/ \
    --clip-norm 20 --max-sentences 8 --max-tokens 12000 \
    --save-dir $EXPDIR/models/ --max-epoch 50 --lr 5e-3 \
    --dropout 0.1 --lr-schedule inverse_sqrt --warmup-updates 4000 \
    --warmup-init-lr 3e-4 --optimizer adam \
    --arch speechconvtransformer_big --distance-penalty log \
    --task translation --audio-input --max-source-positions 1400 \
    --max-target-positions 300 --update-freq 16 \
    --skip-invalid-size-inputs-valid-test \
    --sentence-avg --criterion label_smoothed_cross_entropy \
    --label-smoothing 0.1 &> $logs/train.log
```

Code A.0.4: $\text{E2E}^{\diamondsuit}_{raw}$ Training

Sort References fix

Mattia's script was buggy because when a sentence was skipped, the output file had skipped sentences as well. This script has also an edge case when the skipped line is the last one, but it work for my experiments.

```
import sys

if __name__ == '__main__':
  file = sys.argv[1]
  offset = int(sys.argv[2])
  sents = {}
  with open(file, 'r') as fd:
     for _ in range(offset):
       fd.readline()
     for line in fd:
       if line.startswith('|'):
         continue
       elif line.startswith('S-') or line.startswith('T-') or line.
   startswith('P-'):
         continue
       elif line.startswith('H-'):
         id = int(line.split('\t')[0].split('-')[1])
         sents[id] = line.split('\t')[2]

  for i in range(max(sents.keys()) + 1):
    if i in sents:
      sys.stdout.write(sents[i])
    else:
      print("Error in {}".format(i))
```

Code A.0.5: $FBKFairseqST/scripts/sort-sentences.py fixed

Universitat Politècnica de Catalunya, Barcelona
University of Edinburgh, Edinburgh

2020