Unsupervised Pretraining for Low Resource Spoken Language Translation: Dialect translation

Mohamad Harah

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

There are over 7k languages around the world and most of them are considered to be low-resource, meaning that supervised NLP methods are not effective on them. However, new methods has promised to solve the problem by unsupervised pretraining. We show how SLT can benefit from unsupervised pretraining by running overview comparisons on fine-tuned unsupervised methods as well as deep word-level analysis. we show that unsupervised pretraining is more effective on low resource languages and the improvemnt of infrequent words translation can be double the improvement in frequent words given that they're both in Vocabulary. Furthermore, we find that multilingual pretraining of EN-AR and EN-TA, achieves 28.9 BLEU score for EN-AR, which is the state-of-the-art[*] BLEU result for EN-AR language pair.

[*] To the best of our knowledge we compare results to [Wang et al., 2020] [Aharoni et al., 2019]

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Mohamad Harah*)

Acknowledgements

I'm grateful to God for giving me the strength to complete this project.

I'm grateful to my family and my best friends, at university and overseas, for providing all the love and support that I needed in those tough times.

I'm grateful to my main education supporter, my mum, who will soon be proud to see the first member of the family to graduate university.

I'm grateful to Lexi, my supervisor, for guiding me, for being very patient with me, for believing in me and for providing me all kinds of support throughout the project.

Special thanks to Joshua Wilkins and Vivek Iyer for the technical guidance, the quick responses to answering my questions.

And Special thanks to my best friend Qassem and my two sisters Bdour and Ghena for proofreading my work and the exceptional support.

I Couldn't have done it without any single one of you.

I'm lucky to have all of you.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Research question 2

1.3 Structure of the Report 3

2 Background **4**

2.1 History of Supervised SLT Models 4

2.2 Evaluation Metrics 6

2.2.1 ASR Metrics 6

2.2.2 MT and SLT Metrics 6

2.3 Transformers 7

2.4 Unsupervised Pretraining 7

2.4.1 ASR Unsupervised pretraining (wav2vec) 8

2.4.2 MT Unsupervised Training (mBart) 8

3 Datasets **10**

3.1 Librispeech 10

3.2 Common Crawl Corpus 11

3.3 CoVoST2 11

3.4 IWSLT22 Tunisian Arabic Shared Task 12

4 Experiments **13**

4.1 Data Preprocessing 13

4.1.1 English Audio and Text preprocessing 13

4.1.2 Arabic text preprocessing 14

4.2 Toolkits and Hardware and Setup 14

4.3 Trained/Fine-tuned Models 14

4.3.1 Baseline models 14

4.3.2 Wav2Vec 2.0 15

4.3.3 mBart50 15

4.3.4 MT Analysis 16

4.4 SLT by combining ASR with MT 19

5 Results **20**

5.1 Bilingual and Multilingual fine-tuning 20

6 Discussion **25**
6.1 Results Discussion 25
6.2 Limitations 26

7 Conclusions **27**
7.1 Conclusions 27
7.2 Challenges 27
7.3 Future work 28

Bibliography **29**

A First appendix **32**
A.1 Details about CoVoST2 contents 32

Chapter 1

Introduction

1.1 Motivation

Current supervised Natural Language Processing (NLP) models require a large amount of labelled data to work well, whether their tasks are Automatic Speech Recognition (ASR) or Machine Translation (MT), they require either large amount of transcribed speech in case of ASR, large amount of translated text in case of MT, or large amount of both transcribed and translated speech in case of Spoken Language Translation (SLT) tasks.

There are 7151 languages spoken around the world today and only the top 23 languages are spoken by half of the world's population. This makes the top few languages dominant in resources, while the majority of the rest of languages have significantly fewer resources, resulting in 40% of the world's languages in being considered as endangered [eth, 2022]. This is problematic because the only advantaged languages of current supervised NLP models are the high resource ones, while using the same methods on low resource languages deems to be ineffective.

For languages like Arabic, there are many natively spoken dialects across Arabic-speaking countries. And despite the large number of speakers of these dialects, they have significantly fewer transcription and translation resources compared to Modern Standard Arabic (MSA), which dominates the written and spoken resources. despite MSA being rarely used in interpersonal communications, by being the official language of all Arab League countries, it was easier to spread as the major Arabic Language.

Arabic dialects can vary in similarity to MSA, north-western African dialects like Moroccan and Tunisian have higher deviation from MSA compared to other Arabic dialects. MSA is used in the majority of spoken and written public media and the vast majority of books, articles, newspapers and official communications are done with MSA, which made it relatively rich in transcription and translation resources. It's also used in formal and official communications. This resulted in little attention to dialects, which are the natively spoken languages.

Unsupervised pretraining methods have emerged, those methods were successful in extracting speech context representations from unlabelled speech [Baevski et al., 2020] or text context representation from monolingual corpora [Liu et al., 2020]. This has been proven to allow for cross-lingual learning transfer on speech and text sides. This means we can use models that were pretrained on large unlabelled/monolingual datasets of high resource languages to transfer learning to low resource languages by fine-tuning the pretrained checkpoints on relatively small labelled datasets of transcribed and translated speech.

1.2 Research question

Our main research question is:

How does low resource SLT benefit from unsupervised pretraining?

To answer this, we aim to investigate what the differences are between an ASR-MT pipeline that has been trained on supervised data for a specific language pair, and an ASR-MT pipeline that has been pretrained with unsupervised multilingual data then fine-tuned on supervised data from the same language pair.

We examine the differences between the ASR component pipeline as well as the MT component, and we look at both the high-resource language pair English-Arabic (EN-AR), and its related low-resource language pair English-Tunisian (EN-TA).

In order to determine what advantages the unsupervised multilingual pre-training confers on the low-resource SLT pipeline, we do further analysis on words that are either included or excluded in the machine translation model's fine-tuning vocabulary, and we also examine the differences in frequent vs infrequent words.

We find that unsupervised pretraining can provide significant 28.1 reduction in WER on ASR task for high resource languages like English. We also find that multilingual unsupervised pretraining on MT task can provide 7.5 BLEU improvement for high-resource language pair (EN-AR) and 4.2 BLEU score improvement for low-resource language pair (EN-TA), and we show that the most signifiant improvement is among infrequent words that exist in fine-tuning vocabulary, especially on low-resource languages where improvements among infrequent words were double the improvements of frequent words.

Furthermore, We Experiment with multilingual fine-tuning of English to Arabic and Tunisian Arabic (One-to-many) and compare the results to bilingual fine-tuning of the two language pairs (EN-AR, EN-TA). We find that multilingual fine-tuning of EN-AR and EN-TA improves BLEU score for EN-AR pair by further 2.4 which achieves State-of-The-Art BLEU score of 28.9.

1.3 Structure of the Report

The rest of the report is structured as follows:

- **Background Chapter:** In which we list background knowledge about previous methodologies used for SLT task and their limitations. We also related literature and work on unsupervised pretraining SLT and works with focus on low-resource languages.
- **Datasets Chapter:** Which includes details about all datasets which were used in our training as well as datasets that were used in unsupervised pretraining of large models.
- **Experiments Chapter:** Includes all preprocessing steps, toolkits used, hardware setup, training experiments of our models and methodologies used in analysing results.
- **Results Chapter:** Includes comparisons between all results from training/fine-tuning models, and our findings from analysing those results.
- **Discussion Chapter:** In which we explain our findings and answer our research question, as well as state limitations of our own work.
- **Conclusion Chapter:** Includes a summery conclusion in which we highlight our findings and how they answer our research question. We also list some potential improvement of our work as well as future extensions.

Chapter 2

Background

2.1 History of Supervised SLT Models

There has been a long history of interest in spoken language translation. There has been a workshop on it running since 2004 IWSLT (International Conference on Spoken Language Translation) [Akiba et al., 2004].

Until very recently, such systems were cascade models i.e. a pipeline that combines an Automatic Speech Recognition (ASR) model with a Machine Translation (MT) model [Waibel and Fugen, 2008].

Early ASR models were statistical models, in which audio features can be extracted from raw audio then fed to an acoustic model, which was typically a probabilistic model like Hidden Markov Model (HMM). The acoustic model then maps those features to pronunciations or subwords, which are mapped to words by a pronunciation model. Words are then fed to a language model, which was an N-Gram model that reorders the words to make sentences. Later, HMMs were replaced with Deep Neural Networks (DNNs) in acoustic models. And NGRAMs were replaced with Recurrent Neural Networks (RNNs) while keeping the same high-level structure of the system. An illustration of such system is shown in figure 2.1.

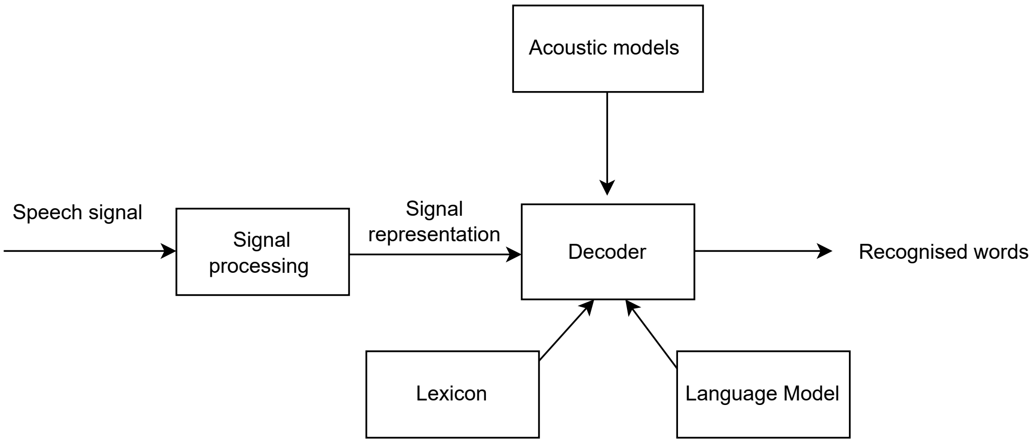

Figure 2.1: High-level representation of early cascade ASR systems with multiple models

When it comes to MT, there were multiple types of systems with different approaches to translation. There were direct systems, which map words and vocabularies directly from a source language to a target language, and there were intra-lingua systems, which take extra steps to perform syntactic analysis and extract semantic features from a sentence from the source language to construct a sentence in the target language from those features. In the early stages, direct systems were very simple and they were outperformed by intra-lingual systems, and as a result, they had lower popularity, that is until the creation of statistical MT models (SMT) [Och and Ney, 2004]. SMT finds the most probable target sentence from a given reference sentence, probabilities are calculated by training a statistical model on a language pair, and it aligns phrases and words automatically. SMT models were competing with intra-lingua models, intra-lingua models had an advantage in constrained domains edge cases, whereas SMT performed better in unconstrained domains cases.

Figure 2.2 illustrates the differences between SLT pipeline with a statistical MT, and another pipeline with an intra-lingua MT model.

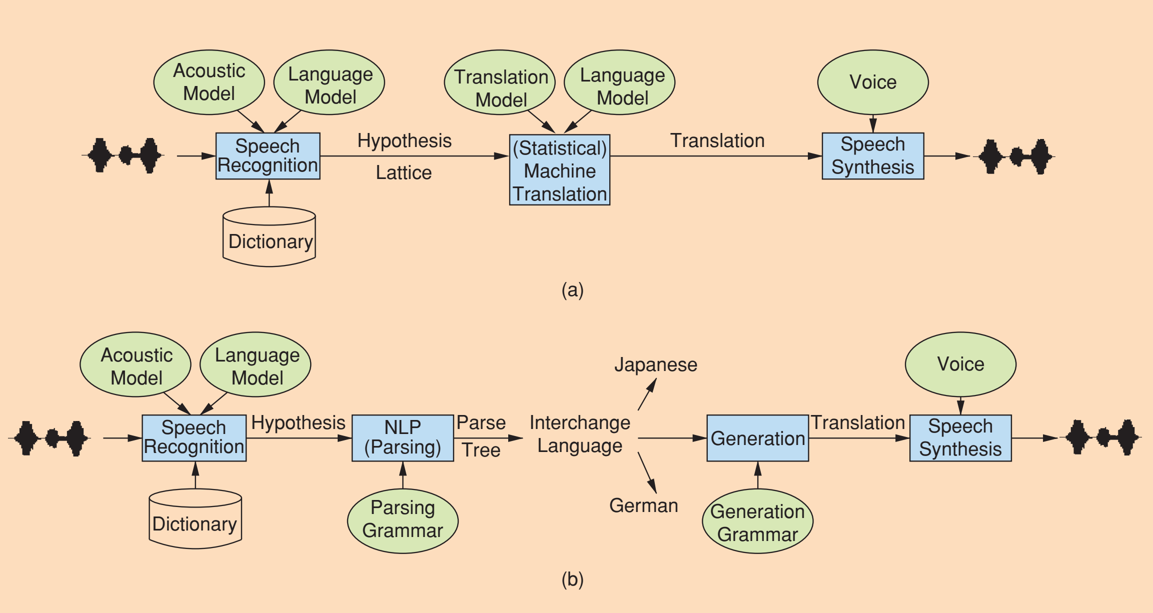

Figure 2.2: High-level representation of (a) SLT pipeline based on SMT model (b) and SLT another SLT pipeline based on an Intra-lingua model, figure from [Waibel and Fugen, 2008]

Limitations of statistical models:
While SMT doesn't require developing grammar rules, but it requires a lot of data (transcripts and reference translations) to be effective, which deemed to be challenging. statical ASR models suffered from the same issue, requiring a lot of data which was difficult to get, especially real conversational speech data. ASR, had also some challenges with noise.

Then recently the advent of neural models has meant that instead of carefully crafting models with many features and then putting them together in a pipeline, end to end systems for ASR [Graves et al., 2013] and MT [Bahdanau et al., 2014] were beginning to show enormous promise.

This led to an interest in an end-to-end model for spoken language translation where you take speech in and output text [Di Gangi et al., 2019].

2.2 Evaluation Metrics

2.2.1 ASR Metrics

WER:

Word Error Rate (WER) is one of the most common metrics used in evaluating ASR models and systems. WER for N words is the number of words added to, substituted or deleted from N. WER is given by the formula:

$$WER = \frac{S+I+D}{N}$$

where S is the number of substitution
I is the number of insertion
D is the number of deletion
and N is the number of the ground truth words.

CER:

Character Error Rate (CER) is another metric that is similar to WER and can be used to evaluate ASR models. It is given by the formula:

$$CER = \frac{S+I+D}{N}$$

where S is the number of substitution
I is the number of insertion
D is the number of deletion
and N is the number of the ground truth characters.

2.2.2 MT and SLT Metrics

BLEU

Bilingual Evaluation Understudy (BLEU) is a score that can be used for NLP Text generation tasks. It's widely used for translation tasks to compare a model predicted translation to one or more reference translations. It was proposed by [Papineni et al., 2002] back in 2002 as an automated method to evaluate translations, replacing the expensive human evaluation methods that were used before. BLEU is calculated by:

$$BLEU = BP.e^{\sum_{n=1}^{N} w_n log p_n}$$

where N is the number of n-grams, the default is 4.
w_n is a weight for each modified precision, the default is $\frac{1}{4}$.

P_n is Modified precision.
And BP is brevity penalty and it's given by:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c \leq r \end{cases}$$

where c is the count of words in candidate/predicted translation.
And r is the count of words in reference translation.

Bleu score can range from 0, where there is complete mismatch between a candidate translation and reference translation, to 1, where a candidate translation is identical to one of the reference translations.

2.3 Transformers

One of the biggest breakthroughs, if not the biggest, in NLP in the past 5 years is the introduction of the Transformer architecture in [Vaswani et al., 2017]. and the exploration of multi-headed attention. Many NLP models started using putting transformers in their architecture, or sometimes parts of the architecture, like BERT using the encoder part of the transformer [Devlin et al., 2018] and GPT-3 using the decoder part [Brown et al., 2020]. The advantage of multi-headed mechanism in transformers is that, it enables the model handle long-range dependencies while solving sequence-to-sequence tasks It examines different parts of the utterance for each predicted label and the model looks predominantly toward previous frames.

2.4 Unsupervised Pretraining

In supervised learning, the model is trained with labelled data, and for the model to achieve effective results, it requires a lot of labelled data. In translation tasks, the labelled data are sentences with reference translations. In ASR tasks, it's a labelled speech and in SLT, the labelled data can be both transcribed and translated speech, which makes it even harder to get enough labelled data for SLT especially if it's for a low resource language.

Unsupervised learning doesn't require data to be labelled at all, meaning that it would only require speech data or monolingual corpora in our case. The advantage of unsupervised method here is that, unlike labelled data, unlabelled data is much easier to collect in large amounts.

Supervised models learn how to predict label Y from data X, while Unsupervised models tends to learn features from its training data, those features can indicate similar and different samples within the data. So the idea behind unsupervised pretraining in NLP is to make models build an understanding by either learning contextual representations from speech like in Wav2Vec2 [Baevski et al., 2020] or develop language understanding like in BERT [Devlin et al., 2018].

Methods for language understanding using language models as unsupervised task by masking out words: [Devlin et al., 2018]

2.4.1 ASR Unsupervised pretraining (wav2vec)

With Unsupervised pre-training ASR models, we aim to learn good speech representations on samples without labels. And then we can fine-tune small, labelled data by masking parts of the sentences and try to make our pre-trained model predict the masked parts.

Wav2Vec is composed of a multi-layer convolutional feature encoder that takes input audio X and outputs latent speech representations Z. Z has then two paths. In the first one, Z is then fed to the Transformer model that outputs the context representations for the entire sequence C. In the second path, Z is quantized from a continoues latent representation to a quantization of discrete units Q. We then set up a prediction task by masking Q onto the transformer model and make the model try to predict some masked quantized units, the process is illustrated in figure 2.3.

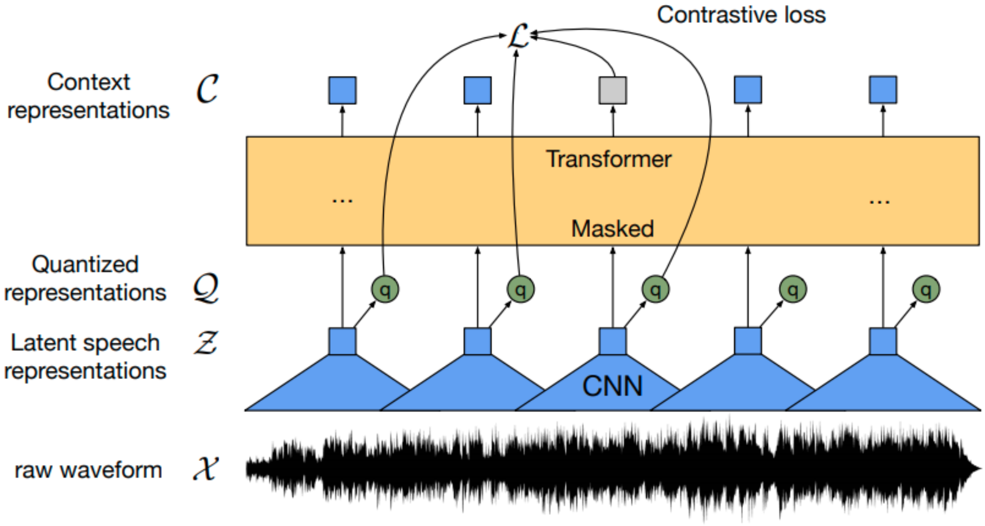

Figure 2.3: Wav2Vec 2.0 Generalized Architecture from [Baevski et al., 2020]

The model can be fine-tuned for speech recognition by adding a single linear projection on top of the transformer model that projects that output representation by the model to a vocabulary (In this case characters). And then train the model with a CTC loss with a low learning rate. A language model can also be added to improve the performance of fine-tuning.

There is also XLSR version of Wav2Vec that is pretrained on multiple languages and can be fine-tuned on small labelled data of any language to achieve impressive results.

2.4.2 MT Unsupervised Training (mBart)

mBart [Liu et al., 2020] is yet another Transformer based model, which was pretrained on monolingual data from multiple languages from CC.25 a subset of Common Corpus corpora. The model is pretrained by receiving masked version of X then by noising function g(x) then it tries to predict the original denoised X. The model can then be fine-tuned in bilingual setting for translation task. mBart promises good performance on unseen data because it doesn’t require the new language alphabet to be in the vocabulary. This makes mBart a very good model for low resource machine translation.

Chapter 3

Datasets

In this chapter, we list all the necessary details about datasets that are either used to pretrain models or we're using to fine-tune and train models.

3.1 Librispeech

Librispeech is a dataset that contains 1000 hours of English speech. The dataset is produced from the LibriVox project, a non-profit project that donates its recordings of audiobooks read by volunteers to the public domain [Kearns, 2014].

The English read speech from audiobooks in Librispeech were automatically aligned and segmented with the corresponding book text, and segments with noisy transcripts were filtered out to produce a corpus of English read speech suitable for training speech recognition systems [Panayotov et al., 2015]. table 3.1 shows details about Librishpeech different subsets.

The pretrained checkpoint of Wav2Vec2 we are using in our experiments (LS-960) is pretrained on 960 hours of unlabelled speech from Librispeech.

subset	hours	per-spk minutes	female spkrs	malespkrs	totalspkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.1	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Table 3.1: Librispeech subsets details

3.2 Common Crawl Corpus

Common Crawl Corpus (CC) is a massive dataset that consists of petabytes of crawled webpages in many languages. [Wenzek et al., 2019] presents a pipeline to create monolingual corpora from CC for more than 100 languages. CC25, a subset of 25 languages out of 100 in CC. CC25, contains varied amount of text from languages from different families. we show the 25 languages with their corresponding number of tokens in figure 3.1.

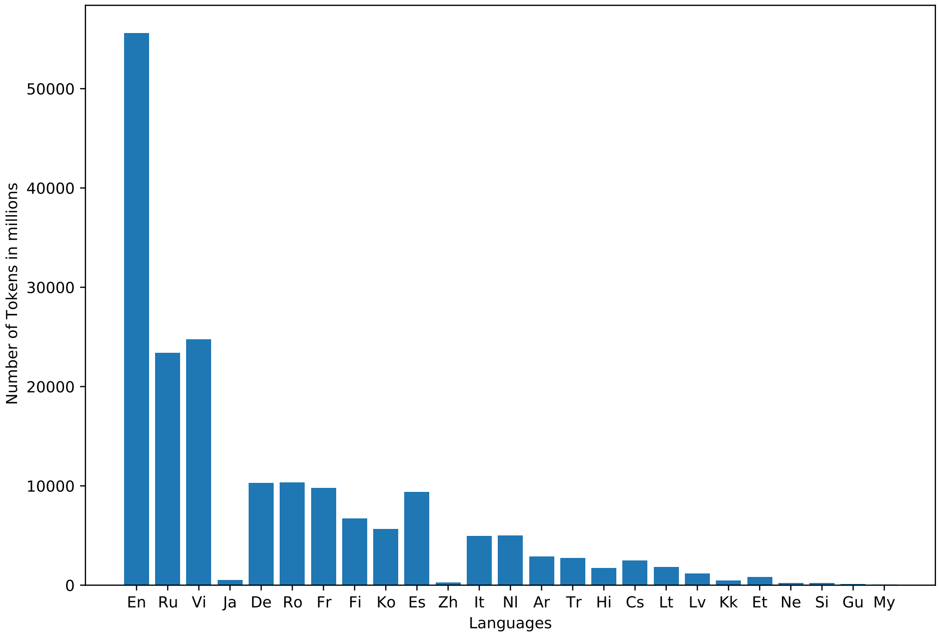

Figure 3.1: CC25 language distribution by the number of tokens in each language (In millions)

The original mBart was pretrained on CC25 [Liu et al., 2020].
mBart50 was pretrained on 50 languages. but instead of pretraining it from scratch, additional monolingual data of 25 languages from [Conneau and Lample, 2019] were pretrained on top of mbart.cc25 checkpoint to extend the languages it is pretrained on to 50.

3.3 CoVoST2

CoVoST2 is a large Speech translation dataset that provides translations from English to into 15 languages and from 21 languages into English. speech and transcription data is originated from Common Voice, a crowd-sourcing project in which volunteers can record sample sentences with their own microphones and review recordings of other users [Ardila et al., 2019]. Translation data is then created from transcripts by professional translators.

we're only interested in language pair EN-AR, which we will train on baselines and fine-tune pretrained models. EN-AR subset consists of (364 train, 26 dev, 25 test) hours of transcribed and translated speech, (10k train, 4k dev, 9k test) speakers and (3M/2M train, 156k/133k dev, 4M/3M test) English/Arabic tokens.

More details about other languages in CoVoST2 are in a snippet from [Wang et al., 2020] in Appendix A.1

3.4 IWSLT22 Tunisian Arabic Shared Task

A Tunisian Arabic speech translation dataset was provided by IWSLT22 [Arrigo et al., 2022]. The Audio consists of conversational telephone speech that was automatically recorded via a robot operating system. The dataset contains 398,064 segments/323 hours of transcribed Tunisian Arabic speech across 206 different speakers and 210,901 segments/167 hours of those are translated into English. After preprocessing and filtering out of sentences that contain non-Arabic letters, the dataset reaches 170k segments. it is then split into 110k train, 30k validation and 30k testing sets to be used as our English to Tunisian (EN-TA) dataset.

Chapter 4

Experiments

In this chapter we include training/fine-tuning steps of our models, starting from data preprocessing. We also mention the toolkits and hardware setup we used. then we mention our analysis strategy and methodology with justifications. Lastly, we show an overall illustration of the combined pipeline.

4.1 Data Preprocessing

4.1.1 English Audio and Text preprocessing

To fine-tune Wav2Vec on CoVoST 2. we filter the 315k EN-AR samples out of the samples with corrupted audio files, we then remove all special characters and punctuations from text because it is hard to classify speech signals to such special characters and punctuations because they don't correspond to sound units, especially without a language model. We keep spaces between words because we need the model to learn to separate words. Otherwise, predictions would always be a sequence of characters, which would make it impossible to separate words from each other. We also keep apostrophes ['] because they can change the meaning of the word when they're added, e.g., "it's" and "its" which have different meanings.

After that, we filter out samples with non-English letters in their English transcript and normalize the text by lowercasing all characters. To make it clear that spaces [" "] have their own tokens, we replace them with more visible words delimiters['|']. We finally get a vocabulary of 30 entries, 26 entries for English alphabet plus apostrophe ['], word delimiter token ['|'], [UNK] and [PAD]. The last two tokens are important for Wav2Vec2 CTC tokenizer. With "unknown" token [UNK] the model can later deal with characters not encountered in CoVoST2 training set and the padding token [PAD] corresponds to CTC's "blank unit", which is a core component of the CTC algorithm [Graves et al., 2006]. Additionally, we filter out the samples with an audio length longer than 5 seconds to save GPU memory by reducing data input size.

The same preprocessing steps were on English transcripts of CoVoST2 (EN-AR) and IWSLT22 (EN-TA) datasets when fine-tuning mBart50 except for replacing spaces. The final number of samples was 290k samples for CoVoST2 after applying all of the above preprocessing.

4.1.2 Arabic text preprocessing

For Arabic and Tunisian transcripts in CoVoST2 and TA-IWSLT22, we apply few simple preprocessing steps. Firstly, we filter out all examples that contains non-Arabic alphabets and non-numerics. Secondly, we filter out diacritics because often, they are not written on words, and their purpose is to assist the reading and pronunciation. Additionally, Arabic words that are identical on character level but with different diacritics and meanings, are uncommon.

4.2 Toolkits and Hardware and Setup

We've used three different toolkits to train our models. Hugging Face was used to fine-tune Wav2Vec2 as well as experimenting with fine-tuning Wav2Vec2 XLSR version, Fairseq was used to train MT models like mBart and baseline tranformer model. Lastly, Speech Brain was used to train CDNN ASR baseline.

We've used various types and GPUs to train and fine-tune our models, the following is a list of what was used for each model, and when training with fairseq, we scale up update frequency for faster training [Ott et al., 2018].

- 2 NVIDIA GTX 1080 were used to fine-tune Wav2Vec2 for English ASR.
- Various numbers of 1 to 8 NVIDIA RTX 3090 were used for all mBart50 fine-tuning experiments.
- 4 to 8 NVIDIA RTX 2080 were used to train baseline models.

4.3 Trained/Fine-tuned Models

4.3.1 Baseline models

Speech Brain CRDNN + CTC (Training ASR from scratch)

We train a CRDNN model from [Ravanelli et al., 2021], which consists of Convolutional Neural Network (CNN) followed by a bidirectional LSTM, which is then followed by a Deep Neural Network (DNN) to output acoustic representations that is given to CTC decoder. first, we tokenize our English transcriptions to unigrams then we train the acoustic model until it converges.

Training MT Transformer from scratch

We train the same MT transformer from [Vaswani et al., 2017] on bilingual EN-AR direction and EN-TA direction. We train with the following hyperparameters: optimizer = Adam, learning rate = 5e-4, dropout = 0.3, weight decay = 1e-4, label smoothing cross entropy with label smoothing = 0.1.

4.3.2 Wav2Vec 2.0

We download Wav2Vec 2.0 base checkpoint pretrained on Librispeech LS-960, and we fine-tune it for ASR task on English using the preprocessed English transcripts of EN-AR split of CoVoST 2.

Wav2Vec2 feature extraction CNN has already been sufficiently trained during pretraining and as claimed by the original Wav2Vec2 paper [Baevski et al., 2020] it does not need to be fine-tuned anymore.

The pretrained Wav2Vec2 checkpoint on Librispeech LS-960 maps a speech signal to a sequence of speech context representations. To fine-tune Wav2Vec2 for English ASR, we add a linear layer on top of the transformer block in figure (2.3). This layer maps a sequence of context representations to its corresponding transcription. And the output size of it corresponds to the number of tokens in the vocabulary, which depends only on the labelled dataset used for fine-tuning and in this case, the vocabulary size CoVoST2 training data is 30.

A pretrained Wav2Vec2 checkpoint expects fine-tuning data to approximately have the same distribution as the data it was pretrained on, meaning both datasets need to have the same sampling rate because the same speech signals sampled at two different rates have a very different distribution, e.g., doubling the sampling rate results in data points being twice as long. Thus, before fine-tuning a pretrained checkpoint of an ASR model, it is crucial to verify that the sampling rate of the data that was used to pretrain the model matches the sampling rate of the dataset used to fine-tune the model.

Wav2Vec2 was pretrained on the audio data of LibriSpeech LS-960 with sampling rate of 16kHz. Our fine-tuning dataset, CoVoST2, was sampled with 48kHz. So, we downsample the speech signal to 16kHz to match the sampling rate of the data used for pretraining.

After processing, tokenizing and extracting features from data, we define a special data collator to pad training batches dynamically meaning, by which all training samples are padded to the longest sample in their batch instead of the overall longest sample in the dataset. This is due to the very large input size in ASR models, e.g., a sample can have an input size of 50000 while it has an output size of no more than 100.

We make use of Hugging Face Trainer to train the model for 30 epochs with learning rate of 1e-4 and weight decay of 5e-3.

4.3.3 mBart50

We explore two different approaches when fine-tuning mBart50. Firstly, we experiment with bilingual fine-tuning of EN-AR direction as well as EN-TA direction. Secondly, we experiment with multilingual fine-tuning one-to-many, that is the direction from EN to AR and TA.

Bilingual fine-tuning: we load mBart50 checkpoint with it's associated dictionary, which contains around 250k tokens from multiple languages. We then tokenize our dataset with Sentencepiece tokenizer. After that, we binarize the data and train with the following hyperparameters: Optimizer = Adam optimizer, Label smoothing = 0.2, Adam episodes = 1e-6 and learning rate = 3e-5 and dropout = 0.3. We apply the above steps for fine-tuning each EN-AR and EN-TA directions. As for hyperparameters, we use the same hyperparameters for both EN-AR and EN-TA with exception to dropout, where in addition to experimenting with dropout of 0.3 we experiment with dropout of 0.1 for EN-TA direction.

Multilingual fine-tuning: we apply the same steps with the same hyperparameters used in bilingual fine-tuning but instead of training on each direction separately, we train on in One-to-Many approach, meaning that we're fine-tuning from English (One) to Arabic and Tunisian Arabic (Many) at the same time.

4.3.4 MT Analysis

We run a some word-level statistical analysis on our MT results to get a better understanding of where the translation improvements of pretrained models over baselines are happening. We focus on two aspects when approaching this problem.

Vocab check:

The first aspect is looking at the existence of words in MT model's fine-tuning vocabulary, i.e. whether the machine learning model has or hasn't seen the word in the fine-tuning set. We calculate the percentage of words that doesn't exist in the fine-tuning vocabulary and compare it to words that exist. There are two ways in which we can approach this. In the first approach, we start from a hypothesis translation and work towards the reference translation. In this case, the score of comparison between two hypothesis translations is translation precision.
The translation precision of a word in hypothesis translation h is given by:

$$Precision_{h \longrightarrow R} = \frac{TP}{TP+FP}$$

Where TP is the number of times a correct translation for the word was found in a reference translation.
FP is the number of times when no correct translation of the word was found in a reference translation.

In the second approach, we start from a reference translation and work towards a hypothesis translation. In this case, the score of comparison between two hypothesis translations is translation recall.

The translation recall for a word in a reference translation R is:

$$recall_{R \longrightarrow h} = \frac{TP}{TP+FN}$$

where TP is the number of times the word in was translated correctly in hypothesis translation h.

FN is the number of times the word was **NOT** translated correctly in hypothesis translation h.

We choose recall as a score of comparison because the recall tells us about how close a hypothesis translation is to predicting the entire words in reference correctly, while precision doesn't regard words that appear in reference translation but don't appear in hypothesis translation.
A sentence-level example: the combined precision of words in the hypothesis translation "welcome home" will report 100% precision on the reference translation "Please, welcome home". While the combined recall for the words in reference translation will report $\frac{2}{3}$ recall.

Our algorithm of determining the correctness or incorrectness of a word translation follows the following simple logic:
For sentence in reference and hypothesis translations, check for each word in reference sentence if it exists in hypothesis translation then increase TP by 1, if it doesn't exist hypothesis then increase FN by 1. Then recall can be calculated with TP and FN values, e.g. the word "world" in reference translation "Hello world" will report a correctness for hypothesis translation "What a peaceful world" (thus increase TP) and an incorrectness for hypothesis translation "Hello people of earth" (thus increase FN).

Frequency Analysis:

The second aspect that we are examining in the analysis is the distribution of words frequencies in the training set with respect to the average translation recall.

First, we need to understand how Arabic and Tunisian Arabic words frequencies are distributed within the training sets. The main focus at this point is how data is split across different unique frequencies. We rank unique words according to their frequency and show them in figure 4.1 and figure 4.2 for Arabic and Tunisian Arabic respectively.

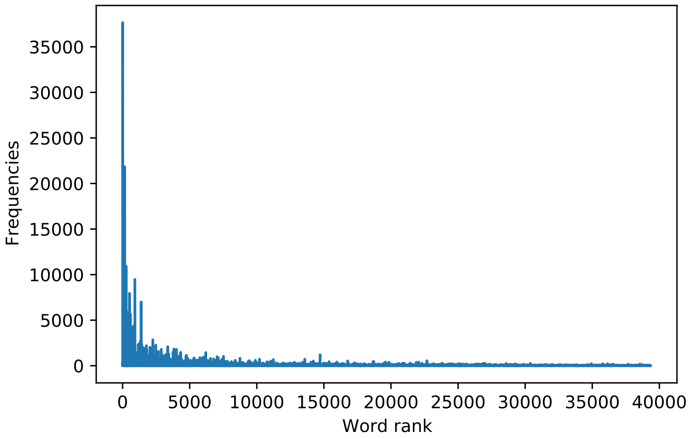

Figure 4.1: Rank of words according to their frequencies in Arabic training set

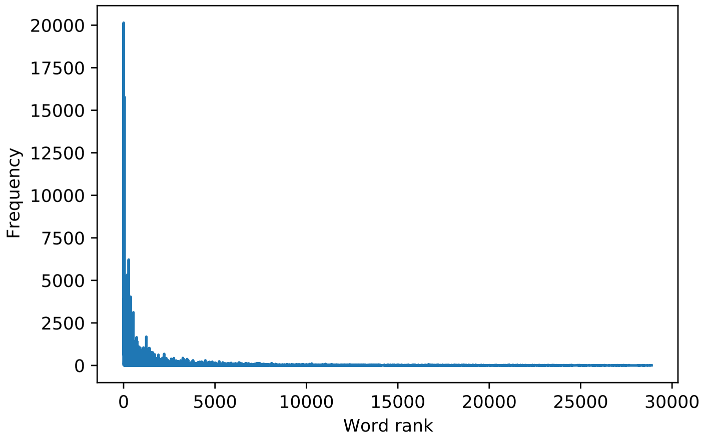

Figure 4.2: Rank of words according to their frequencies in Tunisian training set

As shown in figures 4.1 and 4.2, the top few words have high frequency, but after that, the frequency drops drastically as the rank of the word gets lower.

To make our graph smoother, instead of mapping each word to its frequency, we map each frequency y to the number of words with that frequency x. we get the following figures 4.3 and 4.4 for Arabic and Tunisian Arabic respectively.

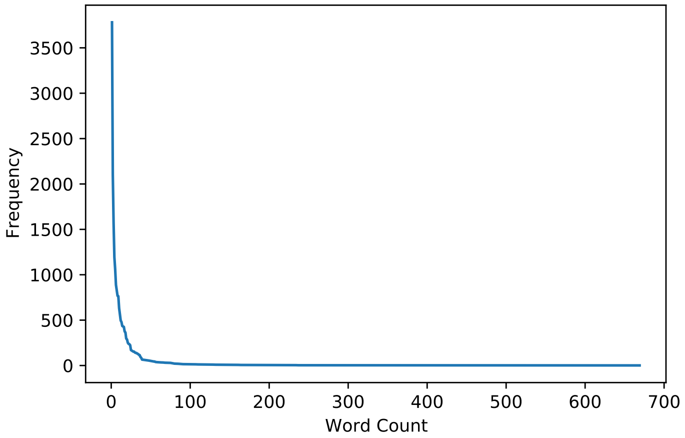

Figure 4.3: Word count per frequency in Arabic training set

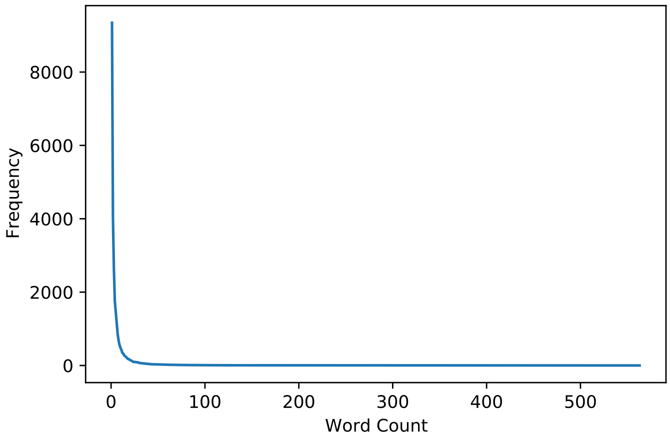

Figure 4.4: Word count per frequency in Tunisian training set

By observing the two figures 4.3 and 4.4, we notice that the graph looks very similar to $f(x) = \frac{1}{x}$ graph. This was expected according to Zipf's Law, which shows that the graph of such words ranking and the probability of appearance of a word P_r is:

$$P_r \cong \frac{const}{r} \longrightarrow f(x) \cong \frac{1}{x}$$

Now that we know how the data is distributed, we group words by unique frequencies and calculate the average translation recall per frequency. We plot a scatter graph of the resulting data, then we fit the data into a polynomial regression model to estimate the average recall at any frequency. (results are in Results chapter)

4.4 SLT by combining ASR with MT

We combine ASR and MT models by mapping the output of Wav2Vec2 to mBart50 preprocessor, where the data is tokenized, binarized and then passed as input to mBart50. The full generation pipeline is illustrated in figure 4.5.

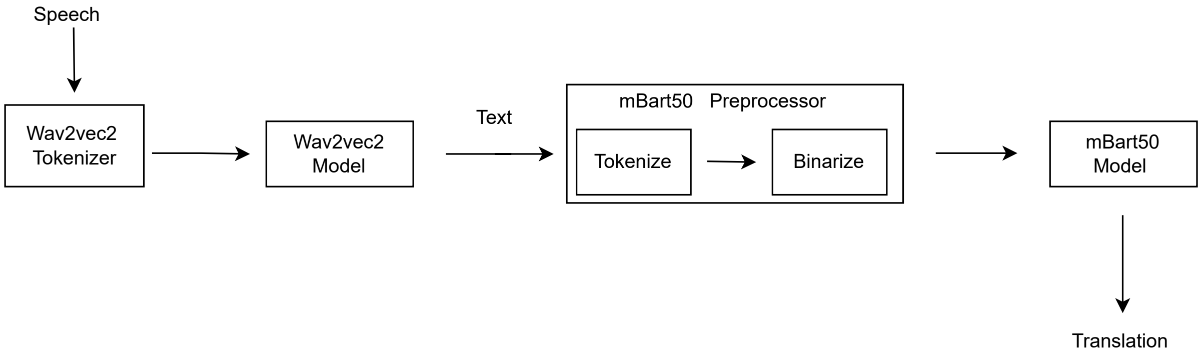

Figure 4.5: full SLT speech-to-text generation pipeline

Chapter 5

Results

In this chapter, we explore the results of our experiments. We break down our main research question into smaller questions. We start by looking at overall comparisons between baselines and fine-tuned pretrained models, and then we experiment with multilingual fine-tuning and compare it to bilingual fine-tuning. And finally, we explore deeper word-level differences to show where improvements of unsupervised pretraining are most significant.

5.1 Bilingual and Multilingual fine-tuning

On ASR side of the pipeline, Table 5.1 presents our findings from fine-tuning Wav2Vec2 for English ASR task compared to training CRDNN on English from scratch.

Model	Validation set	Test set
CRDNN	55	57.3
Wav2Vec2	27	29.2

Table 5.1: WER on the English part of CoVoST2 EN-AR

We can see from table 5.1 that even though both models were trained on the exact same labelled speech, Wav2Vec2 significantly outperforms CDNN by a major 28.1 error difference. This is because Wav2Vec2 was pretrained on large amount of unlabelled speech. This example shows the advantages that pretraining provides, even for high-resource languages.

On MT side of the pipeline, Table 5.2 presents our findings from bilingual and multilingual (One-to-Many) fine-tuning of mBart50 on EN-AR and EN-TA compared to baseline Transformer trained on EN-AR and EN-TA from scratch. We notice an improvement of 7.2 BLEU on EN-AR and 4.8 BLEU improvement on EN-TA with bilingual fine-tuning of mBart50 compared to the baseline. That is a 37% improvement

Language-pair/Direction	Training setting	baseline Transformer	mBart50
EN-AR	Bilingual	19	26.2
EN-AR	Multilingual (One-to-Many)	-	28.9
EN-TA	Bilingual	10.2	15
EN-TA	Multilingual (One-to-Many)	-	13.9

Table 5.2: BLEU score of fine-tuning mBart50 and training basline from scratch on test sets of EN-AR and EN-TA

on EN-AR and a 47% improvement on EN-TA.

	mBart50	Full SLT pipeline	BLEU difference
EN-AR	26.5	11.6	15.1

Table 5.3: BLEU scores of full SLT pipeline, mbart50 and error propagated from ASR part of the pipeline (in BLEU)

In table 5.3 we present The full SLT pipeline performance after combining ASR and MT parts. We compare its BLEU score with BLEU score of MT part of mBart50. We can see that after combining the two models, the pipeline's BLEU decreases by 15.1, which is problematic and will be discussed more in the limitation section of chapter 6.

Language-pair	Model	In Vocab	Not In Vocab	Recall diff
EN-AR	baseline Transformer	0.32	0.17	0.15
EN-AR	mBart50	0.39	0.23	0.16
EN-TA	baseline Transformer	0.17	0.02	0.15
EN-TA	mBart50	0.26	0.06	0.20

Table 5.4: Recall scores for In Vocab and Not In Vocab words, with difference in recall between the two.

In table 5.4 we report the average recall for words that are either included or excluded in the fine-tuning set vocabulary. The (In vocab/Not in vocab) split was about (60% /40%) and (70% / 30%) for AR and TA respectively. And by observing the table above we can see that In Vocab words have more improved results with unsupervised pretraining, especially on low-resource language pair, where the most significant improvement is.

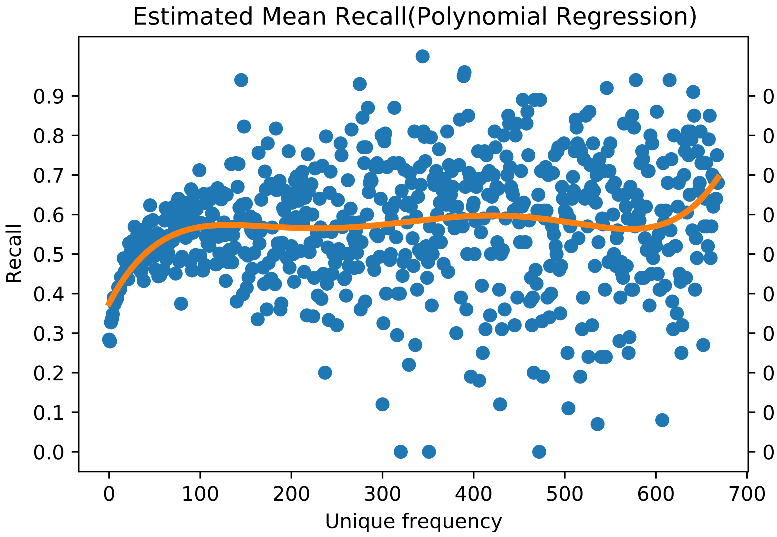

Figure 5.1: (mBart EN-AR) Average recall per unique term frequency scattered and fitted into a 5 degrees polynomial regression model

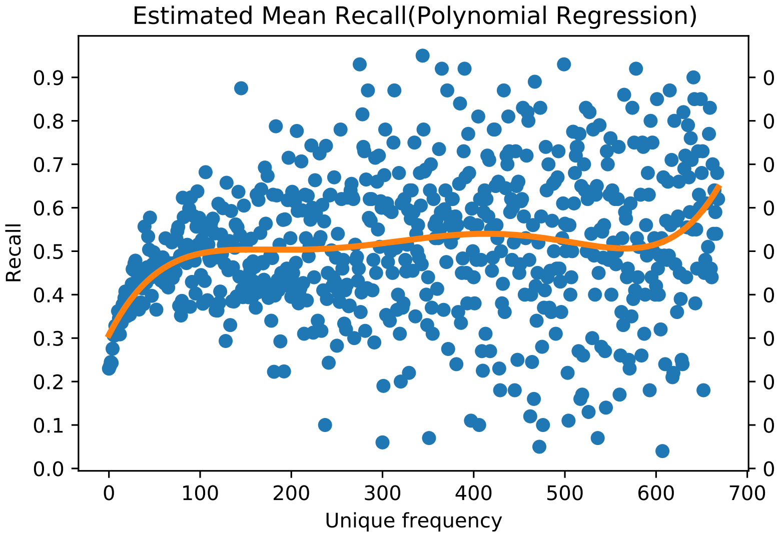

Figure 5.2: (baseline transformer EN-AR) Average recall per unique term frequency scattered and fitted into a 5 degrees polynomial regression model

In figures 5.1 and 5.2 we plot a scatter graph of the average translation recall per frequency for language pair EN-TA on mBart50 and baseline transformer respectively. We fit the data (frequency, recall) into a 5 degrees polynomial regression model to estimate the average recall at any frequency. We notice from scatter plots of both graphs that words with low frequency have consistently lower recall, but as frequency of words increases, the variance in recall values between these frequencies increases as well. To estimate the mean recall per frequency, we fit (frequency, recall) values in a 5 degrees polynomial regression model. From looking at both polynomials, we can see that the difference in recall for each model is only significant between the lowest and the highest frequency ends of the polynomials. In the results of mBart50 model, the average recall starts at around 0.38 for low frequency words, then it climbs to an average of 0.55 recall for mid-frequency words until it reaches 0.7 recall for the highest frequency words. In the results of the baseline transformer model, the average recall starts at around 0.30 for low frequency words, then it climbs to an average of 0.50 recall for mid-frequency words until it reaches about 0.65 recall for the highest frequency words.

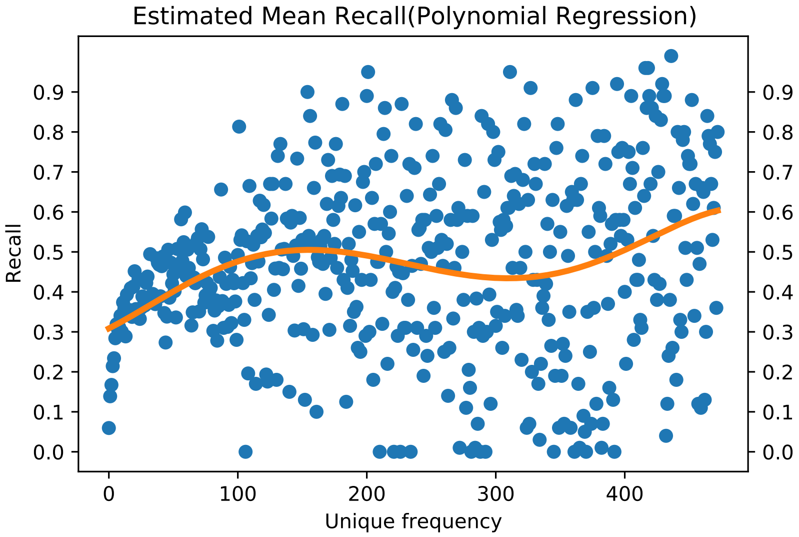

Figure 5.3: (mBart EN-TA) Average recall per unique term frequency scattered and fitted into a 5 degrees polynomial regression model

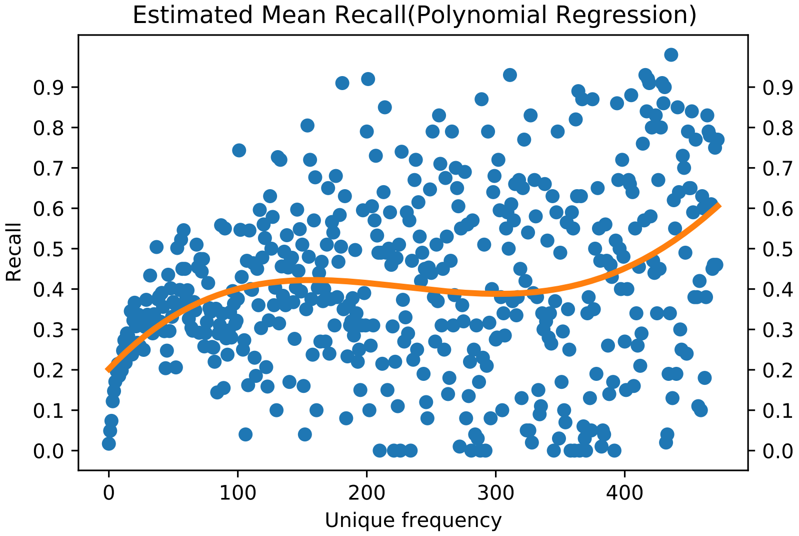

Figure 5.4: (baseline transformer EN-TA) Average recall per unique term frequency scattered and fitted into a 5 degrees polynomial regression model

In figures 5.3 and 5.4 we plot a scatter graph of the average translation recall per frequency for language pair EN-AR on mBart50 and baseline transformer respectively. We fit the data (frequency, recall) into a 5 degrees polynomial regression model to estimate the average recall at any frequency. We notice from scatter plots of both graphs that words with low frequency have consistently lower recall, and similar to the EN-AR pair, as frequency of words increases, the variance in recall values between these frequencies increases as well. However, compared to EN-AR, EN-TA has higher variance on both models (we can see how scatter plots are more spread in higher frequencies in EN-TA pair). Again, from looking at both polynomials, we can see that the difference in recall for each model is only significant between the lowest and the highest frequency ends of the polynomials. In the results of mBart50 model, the average recall starts at around 0.30 for low frequency words, then it reaches an average that varies between 0.45 to 0.5 recall for mid-frequency words until finally it reaches 0.6 recall for the highest frequency words. In the results of the baseline transformer model, the average recall starts at around 0.2 for low frequency words, then it climbs to an average of 0.4 recall for mid-frequency words until it reaches about 0.6 recall for the highest frequency words.

		Average Recall		
Language pair	Words freq	mBart50	baseline Transformer	recall difference
EN-AR	High	0.7	0.65	0.05
EN-AR	Mid	0.55	0.5	0.05
EN-AR	Low	0.38	0.3	0.08
EN-TA	High	0.6	0.6	0.0
EN-TA	Mid	0.45-0.5*	0.4	0.05-0.1*
EN-TA	Low	0.3	0.2	0.1

Table 5.5: Average recall at different different term frequencies, *mid frequency recall is split to high mid freq recall - low mid freq recall

We collect our observed frequency-recall results from previous plots and put them in Table 5.5. Our results show that on EN-AR language pair, mBart50 provides more significant improvements in average recall (0.03) for low frequency words than mid to high frequency words. And on EN-TA language pair, mBart50 provides more significant improvement in average recall (0.1) for low frequency words, which is more than double the improvement in average recall for mid to high frequency words.

Chapter 6

Discussion

In this chapter, we attempt to answer our research question by discussing our findings from our analysis of our experiments results, and list limitations of experiments methodologies and analysis strategies.

6.1 Results Discussion

Our results from ASR training experiments show that unsupervised pretraining can provide a significant 28.1 reduction in WER even for high resource languages like English. We also find that multilingual unsupervised pretraining Improves BLEU scores of high-resource language pair (EN-AR) and low-resource language pair (EN-TA) by 7.5 and 4.8 respectively.

After running some word-level analysis on the improved results, we find that on high-resource pair (EN-AR) translation improvement for words in vocabulary is slightly higher than the improvement for words not in vocabulary. while in low-resource language pair (EN-TA), we find that translation improvement for words in vocabulary are double the improvement of words not in vocabulary.

With further analysis on words frequencies, we show that the improvement in translation, using unsupervised multilingual pretraining, among infrequent words is 60% more than the improvement among frequent words for high-resource language pair (EN-AR), and for low-resource language pair the improvement among infrequent words is double the improvement of frequent words. From the above, we conclude that infrequent vocabulary words in low-resource languages are the top beneficial from unsupervised pretraining.

Finally, our results from multilingual fine-tuning on EN-AR and EN-TA report a 28.9 BLEU score on EN-AR language pair, which, to the best of our knowledge, is the state-of-the-art BLEU score for this language pair. we compare our results to [Wang et al., 2020] [Aharoni et al., 2019].

6.2 Limitations

Our word matching algorithm in our evaluation assumes that a word in a reference sentence is translated correctly if the exact word was found once in the corresponding hypothesis sentence. This doesn't account for words that appear multiple times in either hypothesis or reference sentences. And it also ignores the order of the word. E.g. the word "this" in "this is good apple" would give a correct translation for matching with hypothesis sentences "this this, this this" or "apple good this".

We also use recall score on reference translation exclusively, this means that mistranslated words that exist in hypothesis translation but not in reference translations, will be ignored.

We construct our pipeline from two large pretrained models and we train each one individually. This means that any errors in the ASR model are final and will propagate to the input of the MT part of the pipeline. And no matter how good MT part is, its translation performance will be affected by the ASR model because it wasn't trained on errored data. This is known as error propagation, and new end-to-end solutions try to solve it [Li et al., 2020], [Wang et al., 2021].

Chapter 7

Conclusions

In this chapter, we summarize and conclude our findings and explain how they answer our main research question. Then we reflect on some of the challenges we faced during experimentation. And lastly, we mention some future extensions and potential improvements to our work.

7.1 Conclusions

In this work, we try to find how unsupervised pretraining can help low-resource SLT. We start by looking at overall comparisons between baselines and fine-tuned pretrained models, then we explore deeper word-level analysis to show that the improvement in translation of infrequent words in a low-resource language pair like (EN-TA) can have double the translation improvement of frequent words, given that the words are in the training vocabulary. Furthermore, we show that multilingual fine-tuning on EN-AR and EN-TA can achieve translation performance of 28.9 on EN-AR pair, which is the state-of-the-art BLEU score for this language pair to the best of our knowledge.

7.2 Challenges

The project overall was conceptionally and technically challenging, below we list some of the technical challenges we experienced.

Very large models:
Working with very large models like Wav2Vec2, which has 317 million parameters, and mBart50, which has about 680 million parameters, was very challenging as those models require a lot of memory which is usually only available in the higher end GPUs which we had limited access to. As part of this issue, we were not able to fine-tune Wav2Vec2 XLSR on time to enable fine-tuning on AR-EN and TA-EN language pairs, even though we spent considerable time building it and preparing data for it on Hugging Face.

Training with different toolkits:
Using different toolkits for different models was challenging because each toolkit

requires data to be in a specific shape or processed in a specific way. e.g the low level training scripts with Hugging Face were very helpful to understand how the model works, but it was hard to do multiple experiments with different hyperparameters because any change in input will require a lot of changes in training scripts. Fairseq on the other hand was easier, but it required data in a different shape from Hugging Face. The last toolkit, Speech-Brain was very similar to Fairseq in terms of data preparation.

Filtering Invalid data: there were some invalid data in each of the datasets used, which was sometimes tricky to detect or remove, e.g, there were some invalid audio clips in CoVoST2 dataset which caused tokenizing to fail.

7.3 Future work

An extension to the current project is exploring language pairs AR-EN and TA-EN and run similar analysis to the ones we did in this project to explore the effects of unsupervised pretraining on those two particular language pairs. Such experiments would involve fine-tuning Arabic and Tunisian Arabic Speech on XLSR version of Wav2Vec that was pretrained on unlabelled speech from multiple languages.

We can also do Arabic to Tunisian Arabic and Tunisian Arabic to Arabic with a suitable dataset. In such case, we can multilingually fine-tune EN-AR, EN-TA, AR-TA, TA-AR, AR-EN and TA-EN together to explore and analyse any changes or improvements. Additionally, we can extend dialects to other Arabic dialects or explore entire different low resource languages.

Further analysis on the difference between bilingual fine-tuning and multilingual fine-tuning e.g. discover how did multilingual fine-tuning improved EN-AR BEU score but reduced EN-TA BLEU score.

Finally, with the goal to reduce propagation error from ASR part of the pipeline to the MT part, we can experiment with end-to-end speech-to-text models, which are based on pretrained components like the one in [Liu et al., 2020], and analyse the results of such end-to-ends methods and compare it with our findings from here.

Bibliography

[eth, 2022] (2022). Ethnologue: Languages of the world, how many languages are there in the world? https://www.ethnologue.com/guides/how-many-languages. [Online; accessed 29-March-2022].

[Aharoni et al., 2019] Aharoni, R., Johnson, M., and Firat, O. (2019). Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089*.

[Akiba et al., 2004] Akiba, Y., Federico, M., Kando, N., Nakaiwa, H., Paul, M., and Tsujii, J. (2004). Overview of the iwslt evaluation campaign. In *Proceedings of the First International Workshop on Spoken Language Translation: Evaluation Campaign*.

[Ardila et al., 2019] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., and Weber, G. (2019). Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

[Arrigo et al., 2022] Arrigo, M., Delgado, D., Strassel, S., and Graff, D. (2022). Dialectal speech translation. https://iwslt.org/2022/dialect. [Online; accessed 29-March-2022].

[Baevski et al., 2020] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Conneau and Lample, 2019] Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Di Gangi et al., 2019] Di Gangi, M. A., Negri, M., and Turchi, M. (2019). Adapting transformer to end-to-end spoken language translation. In *INTERSPEECH 2019*, pages 1133–1137. International Speech Communication Association (ISCA).

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

[Graves et al., 2013] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

[Kearns, 2014] Kearns, J. (2014). Librivox: Free public domain audiobooks. *Reference Reviews*.

[Li et al., 2020] Li, X., Wang, C., Tang, Y., Tran, C., Tang, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. (2020). Multilingual speech translation with efficient finetuning of pretrained models. *arXiv preprint arXiv:2010.12829*.

[Liu et al., 2020] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

[Och and Ney, 2004] Och, F. J. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

[Ott et al., 2018] Ott, M., Auli, M., Grangier, D., and Ranzato, M. (2018). Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

[Panayotov et al., 2015] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Ravanelli et al., 2021] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv:2106.04624.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Waibel and Fugen, 2008] Waibel, A. and Fugen, C. (2008). Spoken language translation. *IEEE Signal Processing Magazine*, 25(3):70–79.

[Wang et al., 2020] Wang, C., Wu, A., and Pino, J. (2020). Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*.

[Wang et al., 2021] Wang, C., Wu, A., Pino, J., Baevski, A., Auli, M., and Conneau, A. (2021). Large-scale self-and semi-supervised learning for speech translation. *arXiv preprint arXiv:2104.06678*.

[Wenzek et al., 2019] Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (2019). Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Appendix A

First appendix

A.1 Details about CoVoST2 contents

	Hours (CoVoST ext.)			Speakers (CoVoST ext.)			Src./Tgt. Tokens		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
X→En									
Fr	180(264)	22(23)	23(24)	2K(2K)	2K(2K)	4K(4K)	2M/2M	0.1M/0.1M	0.1M/0.1M
De	119(184)	21(23)	22(120)	1K(1K)	1K(1K)	4K(5K)	1M/1M	0.1M/0.2M	0.8M/0.8M
Es	97(113)	22(22)	23(23)	1K(1K)	2K(2K)	4K(4K)	0.7M/0.8M	0.1M/0.1M	0.1M/0.1M
Ca	81(136)	19(21)	20(25)	557(557)	722(722)	2K(2K)	0.9M/1M	0.1M/0.1M	0.2M/0.2M
It	28(44)	14(15)	15(15)	236(236)	640(640)	2K(2K)	0.3M/0.3M	89K/95K	88K/93K
Ru	16(18)	10(15)	11(14)	8(8)	30(30)	417(417)	0.1M/0.1M	89K/0.1M	81K/0.1M
Zh	10(10)	8(8)	8(8)	22(22)	83(83)	784(784)	0.1M/85K	91K/60K	88K/57K
Pt	7(10)	4(5)	5(6)	2(2)	16(16)	301(301)	67K/68K	27K/28K	34K/34K
Fa	5(49)	5(11)	5(40)	532(545)	854(908)	1K(1K)	0.3M/0.3M	67K/73K	0.2M/0.3M
Et	3(3)	3(3)	3(3)	20(20)	74(74)	135(135)	23K/32K	19K/27K	20K/27K
Mn	3(3)	3(3)	3(3)	4(4)	24(24)	209(209)	20K/23K	19K/22K	18K/20K
Nl	2(7)	2(3)	2(3)	74(74)	144(144)	379(383)	58K/59K	19K/19K	20K/20K
Tr	2(4)	2(2)	2(2)	34(34)	76(76)	324(324)	24K/33K	11K/16K	11K/15K
Ar	2(2)	2(2)	2(2)	6(6)	13(13)	113(113)	10K/13K	9K/11K	8K/10K
Sv	2(2)	1(1)	2(2)	4(4)	7(7)	83(83)	12K/12K	8K/9K	9K/10K
Lv	2(2)	1(1)	2(2)	2(2)	3(3)	54(54)	11K/14K	6K/7K	8K/10K
Sl	2(2)	1(1)	1(1)	2(2)	1(1)	28(28)	11K/13K	3K/4K	2K/2K
Ta	2(2)	1(1)	1(1)	3(3)	2(2)	48(48)	6K/10K	2K/3K	3K/5K
Ja	1(1)	1(1)	1(1)	2(2)	3(3)	37(37)	20K/9K	12K/5K	12K/6K
Id	1(1)	1(1)	1(1)	2(2)	5(5)	44(44)	7K/8K	5K/5K	5K/6K
Cy	1(2)	1(12)	1(16)	135(135)	234(371)	275(597)	11K/10K	79K/76K	0.1M/0.1M
En→X									
De	364(430)	26(27)	25(472)	10K(10K)	4K(4K)	9K(29K)	3M/3M	156K/155K	4M/4M
Tr							3M/2M	156K/125K	4M/2M
Fa							3M/3M	156K/172K	4M/4M
Sv							3M/3M	156K/143K	4M/3M
Mn							3M/3M	156K/144K	4M/3M
Zh							3M/6M	156K/332K	4M/6M
Cy							3M/3M	156K/168K	4M/4M
Ca							3M/3M	156K/171K	4M/4M
Sl							3M/3M	156K/145K	4M/3M
Et							3M/2M	156K/120K	4M/3M
Id							3M/3M	156K/142K	4M/3M
Ar							3M/2M	156K/133K	4M/3M
Ta							3M/2M	156K/121K	4M/3M
Lv							3M/2M	156K/130K	4M/3M
Ja							3M/8M	156K/444K	4M/9M