Maastricht University

Master Thesis

Exploiting multi-lingual data in end to end automatic speech recognition and spoken language translation

Alberto Pérez Robledo

Master Thesis DKE-21-44

Thesis committee:
Dr. Jan Niehues
Dr. Enrique Hortal Quesada

July 28, 2021
Abstract

This thesis investigates how to exploit multilingual data to improve the performance of Automatic Speech Recognition (ASR) and Spoken Language Translation (SLT) models. This approach belongs to the wider scope of techniques working on improving end-to-end performance by optimising the use of available data in deep learning. Furthermore, we also investigate the use of multilingual ASR models to recognise accented speech and reduce the gap in performance between native and non-native speech recognition.

To tackle these issues, various end-to-end multilingual models are trained on top of a self-attention network. The combination of audio and text used to train the models determine the type of model obtained. The baseline models perform monolingual ASR and single direction SLT. The rest of the models perform a combination of these tasks, such as multilingual ASR, multilingual SLT or joint ASR and SLT. Europarl-ST data is used to train the models, which includes audio in 6 languages: English, Spanish, French, German, Italian and Portuguese. The performance of the models is evaluated on various ASR and SLT tasks.

We find the best ASR performance comes from the multilingual ASR models. Low-resource languages in the dataset which train poor monolingual ASR models obtain a much better performance when used to train multilingual ASR models. The average performance improves from 72.9 to 26.5 WER. Single SLT models perform poorly and multilingual SLT models do not improve the SLT performance. However, multitask models trained on joint ASR and SLT perform good SLT. Furthermore, joint training of ASR and multilingual SLT models gives even better performance, improving from 12.8 to 18.6 average BLEU. We believe ASR training in the models creates a good acoustic model. This allows the SLT to use that acoustic model to train correctly on translation.

Multilingual ASR models do not reduce the gap in performance between native and non-native English ASR. Comparing recognition performance on specific accents indicates bilingual ASR models trained on English and the language of the accent perform better than other bilingual ASR models. We believe this is because models trained on a language can improve recognition of that language accent in other speech, but the performance on other accents is harder to determine.
## Contents

1 Introduction .................................................. 3

2 Basic concepts .................................................. 5
   2.1 Supervised learning in ASR and SLT ...................... 5
      2.1.1 ASR and SLT task ................................. 5
      2.1.2 Tasks and mappings ............................... 5
      2.1.3 Supervised learning ............................... 6
      2.1.4 Labeled data in multilingual ASR and SLT .......... 6
   2.2 Neural Networks ............................................ 6
      2.2.1 Design outline .................................... 7
      2.2.2 Core components .................................. 9
   2.3 Encoder-Decoder model .................................... 11
      2.3.1 Sequential inputs ................................ 11
      2.3.2 Auto-regression .................................. 12
      2.3.3 Sequence-to-sequence ............................ 13
      2.3.4 Attention .......................................... 13
   2.4 Transformer ................................................ 14
      2.4.1 Self-Attention .................................... 15
      2.4.2 Query, key and value matrices .................... 15
      2.4.3 Multi-head attention ............................. 16
      2.4.4 Transformer block ................................ 17
      2.4.5 Positional embeddings ............................ 17
      2.4.6 The transformer model ............................. 18
   2.5 Decoding - Beam search .................................. 19
   2.6 Pipeline .................................................... 19
      2.6.1 Preprocessing (text) ............................... 19
      2.6.2 Feature extraction (audio) ......................... 21
   2.7 Evaluation ................................................ 21
      2.7.1 Word Error Rate (WER) ............................ 21
      2.7.2 Bilingual Language evaluation understudy (BLEU) .... 22
3 Related work ........................................... 23
  3.1 Multilingual ASR .................................. 23
  3.2 Multilingual SLT and ASR ......................... 24
  3.3 Accented speech in Multilingual ASR .......... 25

4 Methodology .......................................... 27
  4.1 ASR and SLT performance in multilingual models .... 27
    4.1.1 Models architecture ............................ 27
    4.1.2 Multilingual models ............................ 28
  4.2 Accented speech in multilingual ASR ............... 29
    4.2.1 Test set nationalities .......................... 29
    4.2.2 Additional accented dataset ................... 30

5 Experimental set-up ................................. 31
  5.1 Pipeline ........................................... 31
  5.2 Transformer model ................................ 31
  5.3 Decoding and Evaluation ........................... 32
  5.4 Dataset ............................................ 32
    5.4.1 Europarl-ST .................................... 32
    5.4.2 Additional accented dataset ................... 32

6 Experiments and Results ........................... 33
  6.1 ASR in multilingual setting ....................... 33
    6.1.1 English ASR performance with multilingual ASR ... 33
    6.1.2 English ASR performance with multilingual SLT target-
           ing ASR language (English) ....................... 34
    6.1.3 Low-resource ASR in multilingual setting ........ 35
    6.1.4 Low-resource ASR thresholds for transfer learning in mul-
           tilingual ASR .................................. 36
  6.2 SLT in multilingual setting ....................... 36
    6.2.1 SLT performance in multilingual ASR and SLT .... 36
    6.2.2 SLT improvements given ASR performance .......... 37
    6.2.3 SLT target and ASR language direction ........... 38
    6.2.4 Low-resource SLT in multi-SLT ................... 39
  6.3 Language similarity effects ....................... 40
  6.4 Transfer learning evaluation ...................... 40
  6.5 Accented speech in multilingual ASR .............. 42
    6.5.1 Challenges and assumptions ........................ 43
    6.5.2 Monolingual English ASR performance of native and non-
           native speech ................................ 43
    6.5.3 Improvement of English ASR of native and non-native
           speech in multilingual ASR ....................... 44
    6.5.4 English ASR results by individual nationalities ...... 45
    6.5.5 Additional results by individual nationalities ...... 47

7 Conclusion .......................................... 51
Chapter 1

Introduction

In recent years, thanks to the advent of deep learning, machine performance in automatic speech recognition (ASR) and spoken language translation (SLT) has improved enormously. Speech recognition provides humans with a natural way to interact with machines. It can be used to give instructions without the need of immediate interfaces, such as a keyboard or other gadgets. This removes the need to express thought into other mediums than speech. It also gives an accessible way to record interventions and thoughts. Transcribers are a rare luxury, and having to write while speaking can be cumbersome. In the context of machine learning, speech recognition is known as automatic speech recognition (ASR). Spoken language translation (SLT) is a task related to ASR. In this case, the speech is not only recognised, but also translated. This is useful when a message needs to be communicated in a different language.

Two components are necessary for a deep learning model to perform good ASR or SLT: a learning algorithm and data. Obtaining good quality data can be a daunting challenge, and deep learning models can be very data hungry. Therefore it is important to learn to extract the most information from the available data. A very rich source of data that can be exploited for this tasks is speech from other languages. Speech as a tool has numerous things in common across all languages. Speech is used to communicate thoughts in terms of language. All languages describe and share the same underlying reality. When speech is written down, many languages share similar writing systems. Speech is produced by humans using various speech organs, and the phonetic systems of many languages also have many similarities. A neural network performing ASR/SLT on various languages might improve the performance of any of the individual languages. The other tasks give the network more general informa-
tion about the nature of language and speech processing. The thesis aims to investigate this phenomena.

The second issue related to multilingualism this thesis tackles is the effect of multilingual ASR on non-native English speech recognition. Non-native speakers tend to keep traces from the phonetics of their mother tongue in their speech. This can make accented speech difficult to recognise. Again, a difficult hurdle in this field is the lack of quality data for accented data. This is wide-ranging issue as almost one billion people around the world speak English as a second language (that is more than two times the 350 million who speak it as a first language). Therefore, it is important to ensure ASR system are robust to accent variations. Multilingual ASR training for English ASR could prove a way to reduce accent penalties on performance.

In this thesis we investigate the consequences and advantages of using multilingual data in ASR or SLT. This thesis aims to build on the recent successes of multilingual ASR and end-to-end speech translation. The plan is to create various multilingual models on top of a deep self-attention architecture which can combine ASR and SLT tasks. The performance of these models on ASR and SLT tasks will be compared and evaluated. Furthermore, we explore whether multilingual joint training for English ASR manages to close the accent performance gap. To perform this task we will use the recently released Europarl-ST dataset. This dataset contains audio in English, German, French, Spanish, Portuguese and Italian obtained from the European parliament debates.

The following research questions address the issues that will guide our research:

1. How can we improve ASR in a multilingual setting?

2. How can we improve SLT in a multilingual setting?

3. How can multilingual training help English ASR with non-native accents?

The report has the following structure. Section 2 explains the basics of deep learning and transformers within ASR and SLT context. Section 3 gives context on related work in the field. Section 4 explains the methodology of the different multilingual models. Section 5 explains in more depth the experimental configuration. Section 6 shows the experimental results and Section 7 concludes the report.
Chapter 2

Basic concepts

This section describes how modern automatic speech recognition (ASR) and spoken language translation (SLT) systems work.

2.1 Supervised learning in ASR and SLT

This section defines ASR and SLT more precisely in the context of supervised learning.

2.1.1 ASR and SLT task

In automatic speech recognition an algorithm intakes a sound representation (as the digital format of a recorded sound) and transcribes the speech present in the sound into text using the same language as the speech. The objective is to transcribe the human speech and not other sounds or noises present (at least in the most standard approach). In spoken language translation the objective is similar (to transcribe human speech), however the written form of the speech is in a different language.

2.1.2 Tasks and mappings

Each of these objectives (ASR and SLT) can be considered a task. The term task is a practical way to encapsulate an objective. A few decades ago, recognising speech for different speakers might have been considered different tasks. Nowadays, multi-speaker datasets are common. Talking about tasks allows ex-
plaining the effects of their interactions in terms of transfer learning between different objectives which are more specific.

Both ASR and SLT aim to learning mappings between objects of two spaces. The mappings in both tasks go from audio to text. However, SLT appears as more complex than ASR as the mapping for a translation is more complex than a transcription.

2.1.3 Supervised learning

Supervised learning encompasses a set of methods which learn to map things between two spaces using specific examples of the mapping. The specific examples are used to learn about that mapping in general. The idea is that information about the mapping can be extracted beyond the examples. This is because an assumption exists that the mapping follows some patterns, and there are some abstract relations reflected on the inputs and the matching outputs. The objective is to learn about these relations and then apply it to unseen inputs to predict correct outputs. The process of learning a relation between inputs and outputs is also known as generalisation because the algorithm tries to generalise to the rest of the space from some examples.

2.1.4 Labeled data in multilingual ASR and SLT

Examples used for learning in the context of machine learning are also known as labeled data, where each input is labeled with the correct output. Labeled data in the case of ASR would be the audio of a speech together with the transcribed text. This is similar for SLT except that the transcript would be written in a different language. To be useful, the audios must be split into short segments that are well aligned with their matching text.

2.2 Neural Networks

In this thesis we use a type of neural network known as a transformer to perform supervised learning on the data. The smooth approach to data combination is possible thanks to the design of the algorithm. Before going into the particularities of a transformer, I will explain the standard process neural networks use to learn to generalise from a set of examples.
2.2.1 Design outline

Neural networks are a machine learning algorithm. They use linear algebra combined with easily differentiable non-linear operators to calculate an output for any given input. The aim is for the outputs produced by the network to be similar to the desired outputs provided in the labeled data, so that this behaviour generalises to unseen inputs.

The neural network has an input layer, where the examples it is learning from are encoded into vector or tensor form. This is followed by a series of layers composed of different parameters known as weights. Each layer represents a series of “block” operations by its weights. The operations are usually linear algebra followed by a non-linear operator. These weights are the parameters the network optimises to get the best results. To apply this optimisation, the network first applies this block operations sequentially to the initial input. The intermediate set of weights are known as hidden layers and each hidden layer outputs an intermediate representations which are used in the next layer.

At the end of the network, the final output from the last layer is compared to the golden output from the examples. The comparison is performed through a function known as the loss function. The loss function gives a numeric value on how poorly the network is performing (how “far” the output is from the golden output). Once we have this numeric value for all the examples, we take an average and then use it to optimise the weights. This optimisation uses gradient descent combined with back-propagation to adjust the weights according to the value of the loss. The idea is to apply a correction to its operators so that on the next iteration the outputs will be closer to the desired outputs. Once the loss stabilises during training, the network can be applied to unseen outputs. There is a risk when training the network that the weights will adapt too well to the examples and memorise them. This is called overfitting and might decrease the ability to of the network to generalise to unseen results. To avoid this we use techniques known as regularisation. The most basic one is to add a term that penalises weights for getting too big.

This arrangement effectively creates very powerful search functions for these problem domains. Next, each section of the network is described in more detail.
Figure 2.1: Components neural network. At the bottom, each token receives the embedding before going as input. Then the networks operates on the embedded vectors, giving the hidden layer output and the final layer output. The final layer has a softmax that gives a probability for each token in the vocabulary ($\hat{y}$). The loss is calculated on the probability given for the known correct next token (fish in the image). After doing this for all examples, the loss would be backpropagated (not shown). The task in this network is a language model, but the general process is well illustrated. Image source: Jurafsky and Martin (2020)
2.2.2 Core components

Input Embedding

The neural network takes the training set and accesses the vocabulary of the whole task. The tokens the network will use to break down this vocabulary must be set beforehand. Once it knows the dimension of its vocabulary $|V|$, the network can assign an embedding to each element of the input. This vector embedding is what the network operates on. A basic embedding is to have the network use a $1 \times |V|$ dimension one-hot-vector (of the size of the vocabulary), where all elements are zero, except for the one that indicates the specific word. In a more advanced version, there is an embedding layer, where the network has a weight matrix that modifies the one-hot vector before operating on it. Another possibility is to use an independent dictionary of embeddings.

Feedforward Operation

For simplicity we state the network operates on vectors, however higher dimension elements such as tensors can also work and do not alter the workings of the network. Starting with the input vector, it goes through a matrix multiplication with the weights. This is followed by a non-linear operation, which gives what is known as the activation value. These values become the inputs of the operations in the next layer. This operation continues until the last layer.

The output layer should have the same dimension as the vocabulary. That way each value in the final vector corresponds to one item of the vocabulary. A softmax function is applied over the components of the final vector so as to normalise them. This allows them to be interpreted as probabilities.

Loss function

After operating on the input with all the layers and obtaining the output from the network, a meaningful comparison has to be made to the golden example (true labels). We want a loss function $L(y, \hat{y})$ that describes how close the outputs from the network ($\hat{y}$) are to the true labels ($y$). To motivate the loss function, we have to consider what is the end goal: to get the most number of answers correct. Getting an answer correct corresponds to assigning the highest probability (in the softmax) to the correct word (for the most number of training pairs). In mathematical terms we want to maximise the probability of
getting the correct training label \( y \) given our model \( x \): \( \max p(y|x) \)

For a multinomial classification (where each possible correct token is its own class), the distribution of the correct training label \( y \) given \( x \) looks like the following:

\[
p(y|x) = \prod \hat{y}_k^y_k
\]

This is also known as the conditional expectation. Assuming each label is either true or false, and there is only one correct answer, all the labels will be zero except the correct answer which is one. This is a one-hot vector. The objective is to maximise the above expression, which is equivalent to minimising the negative log: , which is the same a maximising the log:

\[
\text{maximise : } \log(p(y|x)) = \sum y_k \log(\hat{y}_k)
\]

\[
\text{minimise : } -\log(p(y|x)) = \sum y_k \log(\hat{y}_k)
\]

Taking into account that the true labels have a one-hot vector distribution, the final expression to minimise is:

\[
L_{ce}(y, \hat{y}) = -\log(\hat{y}_k)
\]

Where \( \hat{y}_k \) is the output probability from the softmax for the correct label. This expression is also known as the negative log likelihood or loss cross-entropy. Cross entropy can express how difficult it is to encode one distribution in terms of another. In this case how difficult it is to encode the true label distribution in terms of the “estimated” distribution of the network. The lower this difficulty is, the closer this distributions are.

**Gradient descent and backpropagation**

The loss is used to adjust the weight values accordingly through gradient descent. Gradient descent calculates the direction that the gradient is most sharp in and changes the weight values in the opposite direction. The idea is to min-
imise the function by changing the values in the opposite direction of most change. This step requires taking the derivative respect all the operations that produce an output. The neural network has many layers of operations, therefore the process of taking derivatives to update the weights of each layer can be costly. However there is an algorithm known as backpropagation that solves this problem. Backpropagation looks at the network as a computational graph and performs backward differentiation. Keeping track of gradients along the way makes the process more efficient.

2.3 Encoder-Decoder model

The above model explains all of the core mechanics in a neural network, but it has shortcomings when dealing with the sequential data used in ASR and SLT. Speech is a sequence, so to input an utterance into the network, a very long input layer would be necessary to take all the tokens at once. Furthermore, the audio segments may have variable length, which cannot be accounted for unless using a padding technique. One way that the network in Figure 2.1 tries to deal with this using a sliding window approach. However, the tokens outside the window have no impact and the network loses a lot of context.

Some very powerful methods have been developed where tokens go as individual units into the network, and the network has ways to access the rest of the context. This has led to the development of some very capable models known as encoder-decoder.

2.3.1 Sequential inputs

The solution to this problem is to split the input into individual tokens and to modify the network to intake the token inputs individually, using other methods to keep track of the rest of the context. There are two main approaches to this: recurrent neural networks (RNNs) and transformers. Although RNN is not used in our research, it is the most intuitive to explain the encoder-decoder model, and the later advances that led to transformers.

RNN is the most straightforward to understand and was proposed years before the transformer. The RNN takes just one token as input at a time together with a hidden state (from one of the hidden layers) of the network of the previous input. The idea is that as more tokens go in, the hidden state will encode information about them (the network will learn to do this during
Figure 2.2: Each token gets embedded and the goes individually into the network. Unlike in Figure 2.1 where the three tokens went in together. The RNN block visualisation might not make this clear, but each time a token goes in, it is getting processed by the same set of weights as the previous one did, together with the previous hidden state. We can see the network uses the output to generate new outputs by auto-regression training). Therefore the hidden state should contain information about all the previous items in the sequence. Meanwhile, at each time step, the weights that process each token are the same.

An RNN can be represented as a cyclic graph, or unrolled to show the resemblance it has with the standard feedforward model. To backpropagate in this network we have to backpropagate all the way back to the beginning of the sequence for some weights. This is called backpropagation through time.

### 2.3.2 Auto-regression

As this approach processes the inputs token by token, and generates new outputs sequentially, new outputs can be used as inputs into the network. Using an output as an input to generate a new output is a technique known as auto-regression. This property is very useful as we will see, in its most basic form, it can be used to generate output from a prior context.
2.3.3 Sequence-to-sequence

In ASR and SLT the output does not have the same nature as the input. It is a mapping between two objects. The auto-regression approach can be extended to account for this. Instead of using one sequence which keeps generating new output, two sequences are used. They are concatenated with a marker to separate them. The network then learns to extract the information from the first sequence until it sees the marker. From there, it uses the information to auto-regressively generate the second sequence.

In the basic approach the network generates a final hidden state from the first sequence. When inferring, auto-regression begins there and uses that final hidden state to generate the second sequence. We can split these two parts of the network, the first that extracts the information and encodes into a state, and the second one that takes that state and generates the new sequence into encoder and decoder. This is why this set-up is known as encoder-decoder.

2.3.4 Attention

In this situation, the final hidden state plays a big role on the decoder and the output. However, as it stands, this final hidden state might hold more information about recent tokens. There are different approaches to improve the way to hold this information (eg. Bi-RNNs). One powerful method is the concept of attention. As explained above, for each input state in the encoder we get a hidden state that is used for the next input. All these hidden states encode useful information, and as the sequence progresses, some of the early
information might be lost. To use all the information we cannot stack them all together as there is not a fixed number. Similarly, projecting them into one vector might attach more importance to irrelevant aspects. We want to find a way to identify among the hidden states the important information to the needs of the decoder.

In the basic attention approach, a dot product is used to see how relevant each encoder hidden state is to the current hidden state of the decoder. Then they are combined weighted by their relevance. Just using plain dot product for relevance would prioritise the more similar encoder states, however they are not necessarily the most relevant. Therefore a set of weights is added to this dot product attention mechanism, which learns to codify how to identify relevant hidden encoder states.

### 2.4 Transformer

The attention mechanism described in the encoder-decoder section turns out to be very efficient and inspired a very powerful method known as self-attention.

At the moment, I presented the seq-to-seq approach as an extension of the auto-regression that is used with RNNs. However this auto-regression approach will work on any methods where tokens can be processed one-by-one. RNNs need to take in the tokens sequentially (apart from one-by-one), as they need to have the previous hidden state to process it with the token. This means calculations have to be in series and gives problems with gradients having to
backpropagate very far back, or hidden state not holding enough information. These problems can be ameliorated with methods such as ReLUs and LSTMs, but there is a way to avoid the problem altogether.

Attention allows combining many inputs in an efficient way when the number of tokens present is unknown. This way we can work with inputs of any size. Each token in a sequence can be processed individually and in parallel be aware of the rest of the context in the sentence.

2.4.1 Self-Attention

Self-attention applies the attention mechanism to the input sequence itself (this is where the name comes from). In general terms, the mechanism looks at how relevant the other tokens of the sequence are to each individual token. Accordingly self-attention creates a new representation of each token reflecting those relationships. This new representation combines the representations of all the other tokens (including itself). In the next layers, the self-attention operation is applied in the same manner on the new representations of each token (and its relationships). This whole process turns out to give very good results when it comes to sequence-to-sequence problems, and can be adapted very well to the encoder-decoder framework.

2.4.2 Query, key and value matrices

The set-up for self attention is more complex than the basic attention mechanism explained for the encoder-decoder model. Instead of one weight matrix to interpret usefulness, there are three distinct weight attention matrices. These are called query, key and value matrix and are calculated as:

\[
\begin{align*}
q_i &= W_q x_i \\
k_i &= W_k x_i \\
v_i &= W_v x_i
\end{align*}
\]

Each token gets processed individually by the self-attention mechanism. When processing a token we want to find the information most relevant to each token. The query matrix regulates what information each token looks for.
The key matrix regulates what information each token provides to the query. Finally the value matrix regulates the information that each token provides once their relevance has been established by the query and key matrices. As this done individually for each token, without the needs of specific outputs from the other operations, it means it can be parallelised (there are many clones of the self-attention layer performing their respective operations).

The process is quite intuitive for the first layer, as we can think about words being relevant to each other. Each self-attention layer gives an output of identical dimensions for each token, which put together makes an intermediate output. When we apply the self-attention process again, the self-attention is acting on abstract representations, but the mechanics are the same.

### 2.4.3 Multi-head attention

One self-attention mechanism might not be enough, as it might not be able to account for all the factors that the tokens have to pay attention to. Therefore, multiple sets of self-attention (known as heads) can be used. The outputs of
these individual heads are concatenated and projected into one final vector.

2.4.4 Transformer block

There is a small simplification on the explanation above on how layers pass on outputs to each other. In the transformer model, after each self-attention layer there is a feedforward layer and some normalisation. These all make up what is known as a transformer block.

2.4.5 Positional embeddings

Although the attention approach allows to deal with inputs of any size using a token by token approach, getting rid of the sequential aspect means that the temporal aspects are also lost. We bring them back in by using positional embeddings. These embeddings capture inherent relationship between positions (closeness) of the tokens to avoid making the network have to learn what the number indicates. Otherwise, it can be problematic for longer sentences.
2.4.6 The transformer model

The full transformer concatenates the transformer blocks together to create a encoder decoder network. Each block is considered a layer. The encoder uses standard self-attention, however the decoder uses mask self-attention during training. This simulates the way sequential inputs works and blocks the self-attention units from looking at what is next on the text. Figure 2.7 illustrates well the different attentions in the network.

![Diagram of the transformer model](image)

Figure 2.7: Shows where the different attentions take place (explain)
2.5 Decoding - Beam search

To decode at test time, the audio to be transcribed/translated is fed into the encoder. The input from the encoder prompts the decoder to generate output. The decoder generates output sequentially, assigning a probability to every term in the vocabulary at each time step. There are various approaches to select the output from the decoder. The first is to use a locally optimal approach, where at each time step the token with the highest probability is selected. This token is then used by the decoder to generate the next token auto-regressively until an end-of-sequence token is produced. This is known as greedy decoding. This approach is locally optimal, however in the long term it might not give the best result.

Beam search is a technique used to account for subsequent probabilities and obtain more coherent outputs. Instead of selecting the highest probability token at each time step, beam search selects the k-highest ones. Where k is defined by the width of the beam search and determines how many possible outputs are stored in memory. Following this, the k-outputs individually generate auto-regressively their own next output, which gives a softmax with its own probabilities for each of them. The k-best options are selected again. The probability of the path is calculated using the chain rule to select the best combination. The process continues, trimming down to the k-best hypothesis at each time step until the end-of-sequence tokens are reached and the best one is selected. As the path become longer, the products make the total probability more likely to be lower. To avoid penalising long paths, a normalisation method, such as dividing by the length, is applied. Figure 2.8 shows a beam search of width 2.

2.6 Pipeline

The text and the audio have to be adjusted before the transformer processes them.

2.6.1 Preprocessing (text)

Standard preprocessing is applied to the data: a tokenizer (moses), casing (moses) followed by a byte-pair encoder (Seinrich).
Figure 2.8: Beam search of width 2. The branches with the lowest probabilities are dropped. The reason they are negative is because we use logs (of a prob between 0 and 1).

**Tokenizer**  Puts the text into a standard format looking to remove unnecessary constructions from the vocabulary, for example separating words from punctuation marks.

**Casing**  Lower cases words at the beginning of sentences that do not appear capitalised elsewhere in the text.

**Byte pair encoding**  Segments words into subwords, based on the byte pair compression algorithm. The algorithm strikes a balance between a reduced vocabulary size and segments which are not too numerous. It was proposed in the context of machine translation by Sennrich, Haddow, and Birch, 2016 and shown to provide better results for the translation of rare words. Other possible subword segmentations are by character or by n-gram. These subwords are the tokens the networks uses and what the final softmax assigns probabilities to.

Byte pair encoding applies the following algorithm. All the strings are split into their composing characters. The most common pair of elements inside all words are joint together, becoming a new element. The number of times to perform this operation is set by the BPE parameter. If the parameter is left to zero the algorithm returns a character level encoding. On the other hand, if BPE is set to a very high value, it would return the full original vocabulary.
2.6.2 Feature extraction (audio)

Audio is stored in computers as a waveform that has been sampled at a certain rate. To use it in the network it is transformed into a more useful form known as a feature vector. This process has the following steps:

1. Audio is segmented into overlapping windows of 25 ms length. The windows overlap by 10 ms. These windows are called frames and they try to capture specific phonemes. To avoid discontinuities they are smoothed at the edges.

2. Use a discrete Fourier transform to obtain spectral information. We want to know the energy at the different frequency bands which characterises the phoneme.

3. Adjust these frequency bands to human perception properties. Humans distinguish better low-frequency sounds than high-frequency ones. Therefore, the energies are collected according to the mel scale. A unit of pitch in the mel scale corresponds to sounds being perceptually equidistant to a human. A filter bank on the mel scale collects the energies of the frequency bands. Finally, the log is taken to make the values less sensitive to power variations.

4. Last step is to calculate the deltas, which measure the variations of the coefficients between frames and help convey information of change in speech.

The logfbanks values together with their deltas make a vector for each frame which is used as input to our transformer.

2.7 Evaluation

2.7.1 Word Error Rate (WER)

ASR has only one optimal solution (unlike translation). It is unequivocal when recognition is perfect, however evaluating partial correctness can still be difficult. The difficulty lies on how the differences between a partial and a perfect result are assessed. One key word missing might add more value to meaning than three spurious ones.

The standard method used is Word error rate (WER). It assumes all words have the same importance. It compares the hypothesis text to the ref text and
counts the number of mistakes which it categorises into three types: insertions (adding extra word not present in the transcript), deletions (deleting words present in the transcript) and substitutions (word is substituted by a different one). It adds the mistakes up and divides the number by the number of words in the reference. As WER is calculated over a group, the average is weighted by the length of the reference text, so that shorter sentences have less importance.

2.7.2 Bilingual Language evaluation understudy (BLEU)

Unlike in ASR, there is not a perfect output for a translation. Human translators will likely provide different translation for the same sentence. This is because there is no real 1:1 equivalent between languages, and the flexibility and inherent grammatical differences lead to many ways of parsing meaning.

To account for this situation, a flexible metric was proposed by Papineni, Roukos, Ward, and Zhu (2001) 20 years ago. This method compares the output against a golden reference made by a professional translator. The metric calculates how many correct words are obtained at various n-gram levels (i.e. n-words on a row have to be correct to be considered valid). The different n-gram level results are averaged together using a geometric mean and with a penalty for short sentences (BLEU is calculated over an entire corpus).
Chapter 3

Related work

3.1 Multilingual ASR

Traditional ASR systems work by bringing together an acoustic model (AM), a pronunciation model (PM) and a language model (LM). Doing multi-ASR in this systems is challenging (as at least one of the components has to be carefully adjusted to each language we are targeting). End-to-end models such as sequence-to-sequence and CTC approaches avoid having to go through this process. These models are known as language independent.

Watanabe, Hori, and Hershey (2017) are one of the first to show the viability of training many languages together in a language independent set up. They use a hybrid of attention and connectionist temporal classification (CTC) model. They show WER gains respect a language-dependent model using 10 languages (mixture of high and low resource). Toshniwal et al. (2018) work in multi-ASR with 9 Indian languages with much higher resources (around 150 h each). Their ASR model is based on LAS (Chan, Jaitly, Le, & Vinyals, 2015). They show the multilingual model improves WER and that conditioning the encoder and the decoder on the language ID provides more improvements. Zhou, Xu, and Xu (2018) investigate a multilingual set-up using a transformer model (Zhou, Dong, Xu, & Xu, 2018). They use high-resource ASR pre-training on all their models, and the multilingual arrangement gives a 2 point WER improvement. Dalmia, Sanabria, Metze, and Black (2018) also look at multilingual CTC models. They show unrelated high-resource languages help low-resource ones, and performance between monolingual and multilingual matches at 25% of the data in the pre-
trained case. They look into re-training just the encoder, the decoder or the softmax layer and find re-training the whole network gives the best results. Cho et al. (2018) do similar work.

Later work focuses on expanding on these models and seeing the effects of training with more data and more languages. One of the first to use up to 100 languages were Adams, Wiesner, Watanabe, and Yarowsky (2019). They find that when working with new speakers more speech in pre-training gives the best result, over factors such as script, phonological or phonetical similarity or geographical closeness. However this is not the case when adapting to new speech from the same speaker. Although they have many languages, they have around 1500 hours total speech. On average each language has around 15 hours, which is low-resource. Pratap et al. (2020) use around 51 languages, but each with 100 to 1100 hours. The model has 1 billion parameters. Their results show an average WER increase of 23%. To obtain this, they have to increase their model size to 1 billion parameters, as the performance is not as good with smaller models. On top of this they create 6 attention heads with their own decoder for different family groups based on scripts and other factors. This also gives a big improvement, specially for high-resource languages, which otherwise show a degradation in performance. Some recent work by Wang, Pino, and Gu (2020) shows that using SLT as middle task allows for better transfer from high to low resource ASR.

3.2 Multilingual SLT and ASR

Bansal, Kamper, Livescu, Lopez, and Goldwater (2019) investigate the effects of pre-training low-resource SLT on high-resource ASR. They work with end-to-end speech translation from Weiss, Chorowski, Jaitly, Wu, and Chen (2017), which is based on LAS (Chan et al., 2015). They find high-resource ASR improves SLT, and that the ASR does not need to match any of the SLT components. Their ablation study shows the biggest benefit comes from the shared encoder parameters, but, they acknowledge the dependence of the decoder on the encoder makes this assessment unreliable. Stoian, Bansal, and Goldwater (2020) build on these results and find that the WER of the ASR pre-training, and not language relatedness, is the most important factor affecting SLT performance. Bahar, Bieschke, and Ney (2019) compare performance of various ASR and SLT combined architectures together with auxiliary tasks. They look at the
best way to incorporate ASR as pre-training into the SLT task and find that a direct model with and adopter layer gives the best result. Di Gangi, Negri, and Turchi (2019) provide a way to do one-to-many multilingual SLT by using target language embeddings that improves on the source side token approach.

In ASR-SLT task combination, an important issue is how to maximise the information extracted from each task. Le et al. (2020) propose a new dual-decoder transformer architecture which performs both ASR and multi-SLT. It does this by using two decoders. These decoders can attend to the state of the other: the self-attention mechanism uses queries from its own input, but keys and values from the output of the other encoder. Depending on whether it is from the current or previous step, they refer to it as parallel or cross dual decoder. This coupling arrangement achieves SOTA on MUST-C dataset.

Another research branch looks to take advantages of recent advantages of pre-training on unlabelled data. These unlabelled modules usually have a very large number of parameters so new efficient fine-tuning methods are being proposed (X. Li et al., 2021).

### 3.3 Accented speech in Multilingual ASR

Ghorbani and Hansen (2018) experiment with training an end-to-end RNN system on both English and the native language to see if this improves the performance of ASR on English speech with accents from the native language. They look at using the native language for pre-training and multi-task learning. They use two baselines, one trained on just native English, and another trained on native and accented English. They find multi-task learning gives much better improvements than pre-training and improves performance on both baselines. They work with Spanish and Hindi accents, and the systems they train are bilingual (they do not pool both languages together or evaluate on the other accent). Similarly, Vu, Wang, Klose, Mihaylova, and Schultz (2014) look at bilingual and multilingual training of acoustic models for accented speech. They find multilingual and bilingual models perform better than a monolingual model on accented speech. However, bilingual models perform better than multilingual on their matching accent. They look at Bulgarian, Chinese, Indian and German accents.

Most other recent research on accented ASR in end-to-end models focuses on native accents of English dialects (eg Australian vs British English). The reason
for this is that end-to-end models benefit from greater data, and more data is available for this type of accents. Therefore multilingual training is not explored much in end-to-end accented ASR. In this context, mono-dialect training is still costly, so it is common to perform multi-dialect training (Hinsvark et al., 2021). However, native accent training methods are still applicable to non-native training. A common approach is to do multi-task with ASR by pairing it with accent recognition. Adding tokens to the target text indicating the accent of the utterance is an efficient and effective method to do this. B. Li et al. (2017) show this approach gets better performance than pooling all dialects and then dialect fine-tuning. A different approach is to include information about the accents with the input audio itself. One simple way to do this is to use one-hot dialect feature vector together with the input. This approach also performs better than individual dialect models (Grace, Bastani, & Weinstein, 2018) (B. Li et al., 2017). Working with accented speech requires trying to optimise for a specific speaker characteristic (their accent). Therefore it can be placed under the broad task of speaker adaptation (Bell et al., 2021).
Chapter 4

Methodology

4.1 ASR and SLT performance in multilingual models

The aim is to analyse how the performance of ASR and SLT changes when joint training different models on a self-attention network in a multilingual setting.

4.1.1 Models architecture

The work builds on the architecture put forward by Pham et al. (2019). This is a transformer architecture adapted to speech as explained in Section 2. The network uses short audio segments paired with text utterances to perform end-to-end training. To train multilingual models, the network trains on all the language directions in the same way as it would do if it was one language. This means the byte pair encoding is applied over all the languages together. Consequently, a model shares one encoder and decoder for all the language pairings present in training. The softmax operates over all the output languages. Using the same architecture means there is no need to re-train or adapt a decoder. This is a very efficient training method. ASR and SLT use the same architecture and approach. There is no indication to the network or experimental set-up difference between them. This makes the multi-task training efficient to perform. The advantage of this arrangement is that it allows for efficient training. The drawback is that a specific language in a model can only translate to another single language.
4.1.2 Multilingual models

As all the models use the same architecture, changing the languages present in the training set is what creates the different multilingual models. The models are listed as follows:

**Monolingual ASR model**

The model is trained on audio segments from only one source language that targets the text utterances from the same language. These models acts as baselines for the different ASR experiments.

**Bilingual ASR model**

These models are trained using two source audio languages, each paired with text in the same language. These models are effective to assess the impact of ASR from one language on ASR from another.

**Multilingual ASR model**

Trained on many source language audio, each audio is paired with text in its own language. These models show the consequences of using multiple languages in ASR training.

**Single SLT model**

These models act as baselines for SLT performance. Audio from one source language is paired with text from a different language.

**Multilingual SLT model (many-to-one)**

These models are trained on many languages all targeting text in the same language. They show how joint training of SLT tasks targeting the same language affects performance.

**Monolingual ASR with single SLT model (matching or non-matching)**

Two options exist for this type of models. There is one source audio language paired with text from the same language (ASR component), and there is a different audio language which either targets text in the same language as the ASR component (matching) or a different one (non-matching). This model is
the first listed to explore the issue of transfer between different task types. The results can give information on whether a model learns to recognise a language better if the same model can also perform translation from a different unrelated language into the recognition language.

**Monolingual ASR with multilingual SLT**

These models are trained with one source language targeting the same language text (ASR), together with many other source languages all targeting the ASR language. These models allow assessing the impact of many SLTs on one ASR and the impact of one ASR on many SLTs.

**Multi/Bi-lingual ASR with single SLT direction**

These models are trained with two source languages targeting their same target text and an another source language targeting one of the other two targets. They show how the possible effects between the two ASR can relate to the SLT task.

### 4.2 Accented speech in multilingual ASR

English is spoken worldwide as a second language, therefore ASR systems should be tested for use by non-native speakers, many of whom will bring their own accent from their mother tongue. Given this situation, we are interested in seeing the performance of accented data within my English baseline, and if multilingual training has different effects on foreign and native speech. The non-native speech analysis aims to evaluate the performance of the multilingual models on non-native English ASR.

#### 4.2.1 Test set nationalities

The Europarl-ST dataset has the advantage of including many foreign speakers. Using the urls assigned to each speaker in the dataset, each utterance can be labeled with the nationality of its speaker. Subsequently, utterances are grouped into native labels (English or Irish) and non-native labels (all other nationalities). This approach avoids the possible pitfalls of having too few utterances for individual nationalities. The monolingual English ASR model applied to the two groups of speakers can show the performance difference between native and
non-native speakers. The same analysis is performed using multilingual ASR models that included English ASR in training and the performance is compared to a monolingual English ASR model.

Fine-tuning the analysis to individual nationalities would allow analysing how models trained on specific ASR additional directions perform on the accents from specific nationalities. However, there are too many nationalities present in the data to produce enough utterances for each individual nationality. Due to the data limitations regarding each language, the study of this effect on individual nationalities was not feasible.

4.2.2 Additional accented dataset

As the current dataset did not seem capable of providing enough accented data for the languages which the bilingual English ASR models were trained on, a new accented dataset is proposed to overcome the data limitations from Europarl-ST. For this, speakers were selected from specific nationalities in the EU parliament site which had interventions in English. The speakers had to speak as first language one from the dataset (Spanish, French, German, Italian, Portuguese). The English transcripts for the selected speeches were split into shorter lengths to work as text utterances. Following this, we listened to each intervention and noted down the time delimiters where each utterance started and finished. This non-native speech analysis uses the bilingual ASR models together with the multilingual ASR models to compare performance on individual accents. The objective is to trace more closely the influence of the individual ASRs on English ASR performance.
Chapter 5

Experimental set-up

5.1 Pipeline

The Moses toolkit\footnote{http://www.statmt.org/moses/}\footnote{https://github.com/moses-smt/mosesdecoder} is used to tokenise and apply the correct casing to the text utterances. This is followed by Byte Pair Encoding\footnote{https://github.com/rsennrich/subword-nmt} from Sennrich et al. (2016) set to 10,000 merges. BPE is applied jointly over all the languages a model is trained on. The Python Librosa library\footnote{https://github.com/librosa/librosa} is used for feature extraction. The logfbanks and the deltas have dimensions of 26 each, making a total size for the feature vectors of 52.

5.2 Transformer model

The model architecture follows Pham et al. (2019). A down sampling method is used to reduce the length of input vectors by concatenating 4 feature vectors together. The encoder depth is set to 12 layers and the decoder to 4. Experiments performed with 12 layers for the decoder give very similar performance, so a 4 layer depth is chosen for a 25\% shorter training time. Attention dropout is set to 0.2, dropout to 0.2, learning rate to 0.001. The model size is 512, it uses stochastic layers, 8 heads and the optimiser is Adam (Kingma & Ba, 2017). The model also uses stochastic layers (Pham et al., 2019) to avoid overfitting. Hyperparameters are kept constant for all the models.
5.3 Decoding and Evaluation

Beam search is set to one. The WER 5 and the BLEU 6 are calculated using tokenised case sensitive format. Experiments showed beam set to 8 gave an overall 2% WER gain, however calculating WER on untokenised case sensitive resulted on similar 2% loss of performance.

5.4 Dataset

5.4.1 Europarl-ST

Europarl-ST (v1) (Iranzo-Sánchez et al., 2020) has speech in 6 languages (English, Spanish, French, German, Italian, Portuguese) which are transcribed and translated in all directions. The amount of training data 7 is summarised in Figure 5.1

<table>
<thead>
<tr>
<th>src/tgt</th>
<th>en</th>
<th>fr</th>
<th>de</th>
<th>it</th>
<th>es</th>
<th>pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>-</td>
<td>81</td>
<td>83</td>
<td>80</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>fr</td>
<td>32</td>
<td>-</td>
<td>21</td>
<td>20</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>de</td>
<td>30</td>
<td>18</td>
<td>-</td>
<td>17</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>it</td>
<td>37</td>
<td>21</td>
<td>21</td>
<td>-</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>es</td>
<td>22</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td>pt</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 5.1: Hours of training data

5.4.2 Additional accented dataset

In total, 150 extra utterances are gathered from 12 speeches from 6 speakers (2 Spanish, 2 French, 2 German). These are obtained from the European Parliament site 8. The method is explained in Section 4.2.2.

---

5https://github.com/jitsi/jiwer
6https://github.com/mjpost/sacreBLEU
7https://www.mllp.upv.es/europarl-st/
6.1 ASR in multilingual setting

In this section we compare the ASR performance of different models. Sections 6.1.1 and 6.1.2 discuss English ASR evaluation because the English monolingual ASR model obtains good performance. Sections 6.1.3 and 6.1.4 examine the effects on the other languages which have poor monolingual ASR models.

6.1.1 English ASR performance with multilingual ASR

The first results show that English performance improves when joint training with other languages in multilingual ASR. Table 6.1 shows the improvement increases as more languages and more utterances are used in training. The monolingual English ASR has a baseline score of 26 WER, training with 31,500 utterances. The best score comes from the model trained on 5 extra ASR directions (49,000 utterances) at 22.4 WER. This is a 3.5 point improvement.

For the bilingual ASR models, the improvements do not vary much depending on the language, and appear more correlated to the number of utterances. The improvements of the multilingual ASR models do not increase linearly. A WER improvement from 26 to 24.5 (-1.5) requires around 12,000 utterances, while from 22.7 to 22.4 (-0.3) requires 16,000 utterances. This non-linearity is expected, as most deep learning systems become more data hungry as performance improves.
<table>
<thead>
<tr>
<th>Language (Audio - Text)</th>
<th>WER</th>
<th>Addition</th>
<th>Utterances English</th>
<th>Non-English</th>
</tr>
</thead>
<tbody>
<tr>
<td>En - En</td>
<td>25.95</td>
<td>Baseline</td>
<td>31607</td>
<td>0</td>
</tr>
<tr>
<td>En,Es - En,Es</td>
<td>24.66</td>
<td>+1 ASR</td>
<td>31607</td>
<td>7402</td>
</tr>
<tr>
<td>En,Fr - En,Fr</td>
<td>24.19</td>
<td>31607</td>
<td>12446</td>
<td></td>
</tr>
<tr>
<td>En,De - En,De</td>
<td>24.46</td>
<td>31607</td>
<td>12904</td>
<td></td>
</tr>
<tr>
<td>En,Es,Fr,De - ASR</td>
<td>22.71</td>
<td>+3 ASR</td>
<td>31607</td>
<td>32752</td>
</tr>
<tr>
<td>En,Es,Fr,De,It,Pt - ASR</td>
<td>22.38</td>
<td>+5 ASR</td>
<td>31607</td>
<td>48955</td>
</tr>
</tbody>
</table>

Table 6.1: Multilingual ASR with English baseline. As ASR directions are added to the English baseline, the WER decreases.

### 6.1.2 English ASR performance with multilingual SLT targeting ASR language (English)

This set of experiments evaluate the performance when training the ASR together with other languages translating into the ASR language. The question addressed is whether learning to translate to a language will give the network more information about recognising that language. The results show models trained on joint ASR and SLT provide a considerable lower improvement than multilingual ASR models when using the same number of additional utterances. Table 6.2 shows a single additional SLT task in training provides no improvements, and neither do 3 SLT tasks, where the WER remains at the same value (26) than the baseline monolingual ASR. Training with 5 SLTs does give some improvement of around 1.2 points. This improvement is slightly less than the given by the Spanish bilingual ASR model as shown in Table 6.1, however it requires 7 times more utterances (∼7,000 vs ∼50,000). The network is learning to translate, as the BLEU is above 10 for all of the SLT tasks, however there seems to be little transfer taking place from SLT to ASR.
<table>
<thead>
<tr>
<th>Language (Audio - Text)</th>
<th>WER</th>
<th>Addition</th>
<th>Utterances</th>
<th>English</th>
<th>Non-English</th>
</tr>
</thead>
<tbody>
<tr>
<td>En - En</td>
<td>25.95</td>
<td>Baseline</td>
<td>31607</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>En,Es - En</td>
<td>26.07</td>
<td>+1 SLT</td>
<td>31607</td>
<td>7402</td>
<td></td>
</tr>
<tr>
<td>En,Es,Fr,De-En</td>
<td>26.17</td>
<td>+3 SLT</td>
<td>31607</td>
<td>32752</td>
<td></td>
</tr>
<tr>
<td>En,Es,Fr,De,Pt,It- SLT:En</td>
<td>24.70</td>
<td>+5 SLT</td>
<td>31607</td>
<td>48955</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Multilingual SLT targeting English baseline. Many SLT directions provide a small improvements on baseline WER.

### 6.1.3 Low-resource ASR in multilingual setting

The monolingual ASR models give clear performance differences for different amounts of utterances. English ASR model with a high amount of utterances obtains a WER of 26. Spanish, French, German, Italian and Portuguese train on less than 13,000 utterances and all their WERs are higher than 50. Therefore these 5 languages can be considered low-resource.

The multilingual ASR model shows that performance of the low-resource languages improved by far more than the high-resource English ASR. English ASR achieved a best WER of 22.7 in the multilingual ASR model, which was around a 3 points (12%) total WER improvement over its monolingual model. On the other hand, Spanish ASR lowered to 29.4 (79 points), French ASR to 28.4 (34 points) and German ASR to 29.10 (36 points). The average performance improves from 72.9 to 26.5 WER. The results are shown in Table 6.3. This result is understandable as the low-resource languages get more auxiliary utterances. More importantly, their baselines have very poor WER scores (above 50) which provide a big margin for improvement.

To investigate whether the cause for this improvement was a transfer from the high-resource ASR to the low resource ASR, another multilingual ASR model was trained were all the ASR components were low-resource languages. Table 6.3 shows that combining many low-resource gives also good WER performance for all languages. This indicates a high-resource language is not necessary to boost low-resource ASR performance.
Table 6.3: Low-resource performance. Monolingual experiments give poor results. However combination with a high-resource ASR (second row) or many low-resource ASRs (fourth row) provide big improvements in performance.

### 6.1.4 Low-resource ASR thresholds for transfer learning in multilingual ASR

The high improvements of low-resource ASR in multilingual ASR training indicate that it might be possible for even lower utterance counts ASR components to benefit from transfer learning. This could open the possibility to few-shot ASR. To evaluate this, the size of one of the ASR components is further reduced in a multilingual ASR model. The results show that the performance decreases considerably in this very low data situation. Table 6.4 shows that with 2,000 utterances the performance clearly degrades and it worsens even further at 500 utterances (although it is still better than the full low-resource monolingual).

<table>
<thead>
<tr>
<th>Low-resource component</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fr (2000)</td>
<td>43</td>
</tr>
<tr>
<td>Es (500)</td>
<td>53</td>
</tr>
</tbody>
</table>

Table 6.4: Artificially reduced components in a many low-resource experiment to test lower bounds ASR. Performance degrades for very low number of utterances.

### 6.2 SLT in multilingual setting

In this Section we compare the SLT performance of different models.

#### 6.2.1 SLT performance in multilingual ASR and SLT

Unlike in the ASR models, both of the single SLT models (Spanish to English and English to Spanish) perform poorly. This can be seen in Table 6.5 and Table 6.6. The first baseline has 8,000 utterances and the second has 31,500. Table 6.5
shows adding more SLTs to the training which target the same language alters the score negligibly. In the 5 SLTs model there are a total of 50,000 utterances and the BLEU for Spanish to English is just 2.4.

On the other hand, adding ASR in the same direction as the SLT target improves the SLT performance considerably. Table 6.5 shows that the SLT task score increases from 2 to 11.7 BLEU with an additional ASR direction. Furthermore, ASR has the effect of allowing multilingual SLT to help as well. Adding another 4 SLT directions increases the BLEU score from 11.7 to 18.5.

<table>
<thead>
<tr>
<th>Language (Audio - Text)</th>
<th>BLEU</th>
<th>Addition</th>
<th>Utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Es - En</td>
<td>2</td>
<td>Baseline</td>
<td>7402</td>
</tr>
<tr>
<td>Es,Fr,De,It,Pt - En,<em>,</em>,<em>,</em></td>
<td>2.4</td>
<td>+4 SLT</td>
<td>48955</td>
</tr>
<tr>
<td>En,Es - En,<em>,</em></td>
<td>11.7</td>
<td>+1 ASR</td>
<td>7402</td>
</tr>
<tr>
<td>En,Es,Fr,De,Pt,It - En,<em>,</em>,<em>,</em>,_</td>
<td>18.5</td>
<td>+1 ASR, + 4 SLT</td>
<td>48955</td>
</tr>
</tbody>
</table>

Table 6.5: SLT Es-En baseline with additional ASR to SLT target. ASR improves sharply the SLT performance. ASR also makes the additional SLT tasks to the same target provide a considerable improvement (+ 6.8 BLEU)

### 6.2.2 SLT improvements given ASR performance

The previous experiment shows ASR training is crucial for a model to obtain a good SLT performance. To look further into this effect we compare the performance of SLT for different ASR quantities. One model trains on a high-resource (low WER) auxiliary ASR, and the other trains on a low-resource (high WER) ASR. Table 6.6 shows the English to Spanish baseline (31607 utterances) on the left side and the Spanish to English baseline (7402 utterances) on the right side.

At the baseline level, the model trained on more utterances has a slightly higher BLEU, but both are very low (3.6 and 2). A Model trained with an additional ASR component shows that the Spanish to English SLT performance increases to 11.7 (this is the same result presented in the previous section). However, the English to Spanish SLT performance does not see such an improvement when training on a low-resource ASR component, even when the SLT has 4 times the number of utterances than the other model.

This difference in SLT performance with high and low ASR remains when
using more than one SLT. Multilingual SLT improves considerably the score of the single SLT if high-resource ASR is present in training. On the other hand, the low resource ASR model gets worse performance for SLT. This might occur because the ASR is also getting worse (WER of 101.66) as the ASR examples are diluted among the other SLT training tasks.

A possible explanation for this behaviour is that the additional ASR in the training, which allows the SLT to perform well, is not effective enough in the English to Spanish case. The WER of the Spanish ASR in the model is 87.88, which entails a very high error. On the other hand, for Spanish to English, the WER of the English ASR in the model is 26.1, which is much lower.

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>WER</th>
<th>Utt.</th>
<th>Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>En - Es</td>
<td>3.6</td>
<td>-</td>
<td>0</td>
<td>Baseline</td>
</tr>
<tr>
<td>Es,En - Es,En,Fr,De - SLT (Es)</td>
<td>4.8</td>
<td>87.88</td>
<td>7402</td>
<td>+1 ASR</td>
</tr>
<tr>
<td>Es - ASR, En,Fr,De - SLT (Es)</td>
<td>3.1</td>
<td>101.86</td>
<td>7402</td>
<td>+1 ASR, 3 SLT</td>
</tr>
<tr>
<td>Model</td>
<td>BLEU</td>
<td>WER</td>
<td>Utt.</td>
<td>Addition</td>
</tr>
<tr>
<td>Es - En</td>
<td>2</td>
<td>-</td>
<td>0</td>
<td>Baseline</td>
</tr>
<tr>
<td>En,Es - En,En,Fr,De - SLT (Es)</td>
<td>11.7</td>
<td>26.07</td>
<td>31607</td>
<td>+1 ASR</td>
</tr>
<tr>
<td>Es - ASR, En,Fr,De - SLT (En)</td>
<td>13.3</td>
<td>26.17</td>
<td>31607</td>
<td>+1 ASR, 3 SLT</td>
</tr>
</tbody>
</table>

Table 6.6: Difference in performance of SLT task with low wer ASR (left side) and high wer ASR (right side).

### 6.2.3 SLT target and ASR language direction

We hypothesise that the ASR component matching the SLT target in training gives more information to the model about the target. This model learns more target words and structures from the ASR, which helps translation. An additional experiment is set-up to test this. One model is trained on a low-resource ASR direction (Fr-Fr) matching the SLT target (Es-Fr). Under normal conditions this does not help as shown in the section above. However, an additional high-resource ASR (En-En) is also included in the training set-up, which does not match the SLT target (Fr). Theoretically, this high-resource component should improve the score of the low-resource French ASR, and consequently improve the BLEU score. The second model just has a SLT task (Es-Fr) and a high-resource non-matching ASR direction, therefore there is no language rela-
tion between the ASR and the SLT.

Table 6.7 (final 2 rows) shows both models produce good BLEU scores. This was expected for the model with an intermediate ASR component, where the low-resource ASR did obtain a good WER score (29.8) which allowed SLT to perform well (15.8). The good BLEU score from the second experiment falsifies our hypothesis and shows the ASR language and the SLT target can be different. It seems using the same language is not what causes this increase in performance. Furthermore, even though the model with a matching SLT and ASR scores a slightly higher BLEU (15.8), the result might be caused by the higher amount of total ASR utterances in training producing a lower WER score.

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>WER</th>
<th>Addition</th>
<th>Utt.</th>
<th>SLT</th>
<th>ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High match ASR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Es,Fr,De,It,Pt – SLT:En</td>
<td>2.4</td>
<td>-</td>
<td>Baseline:</td>
<td>48955</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>En - ASR,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Es,Fr,De,It,Pt – SLT:En</td>
<td>18.5</td>
<td>26.2</td>
<td>+1 ASR</td>
<td>48955</td>
<td>31607</td>
<td></td>
</tr>
<tr>
<td><strong>Low match ASR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fr - ASR,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>En,Es,De – SLT:Fr</td>
<td>4</td>
<td>90.4</td>
<td>Baseline:</td>
<td>18582</td>
<td>12446</td>
<td></td>
</tr>
<tr>
<td>En,Fr - ASR,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Es,De,Pt,It – SLT:Fr</td>
<td>15.8</td>
<td>29.8</td>
<td>24.4</td>
<td>21723</td>
<td>31607,12446</td>
<td></td>
</tr>
<tr>
<td><strong>No match ASR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>En - ASR,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Es,De,Pt,It – SLT:Fr</td>
<td>14</td>
<td>-</td>
<td>25.5</td>
<td>21723</td>
<td>31607</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.7: SLT performance with different kinds of matches between SLT target and additional ASR

### 6.2.4 Low-resource SLT in multi-SLT

The SLT performance benefits greatly from the multilingual SLT in training once ASR is present, as shown in Table 6.8. Unlike in the ASR joint training, there is no high-resource SLT in the data. All SLT tasks require an ASR auxiliary component to ensure good performance. Average performance improves from
12.8 to 18.6 BLEU.

<table>
<thead>
<tr>
<th>Model</th>
<th>Es-En</th>
<th>Fr-En</th>
<th>De-En</th>
<th>It-En</th>
<th>Pt-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual (+ En ASR)</td>
<td>11.7</td>
<td>16.1</td>
<td>10.1</td>
<td>16.2</td>
<td>9.80</td>
</tr>
<tr>
<td>Multi SLT (+ En ASR)</td>
<td>18.5</td>
<td>21.3</td>
<td>14.4</td>
<td>19.7</td>
<td>19.2</td>
</tr>
<tr>
<td>Utterances</td>
<td>7402</td>
<td>12446</td>
<td>12904</td>
<td>11285</td>
<td>4918</td>
</tr>
</tbody>
</table>

Table 6.8: Joint training of the SLT tasks (second row) gives a considerable performance improvement for all the the SLT scores

## 6.3 Language similarity effects

This experiment aims at assessing if language similarity can affect the transfer learning between languages in multilingual ASR and SLT. New models are trained for two arrangements were transfer learning has been observed (multilingual ASR and multilingual SLT). For each task, Spanish is grouped with a set of related languages (Portuguese and Italian) and non-related languages (French and German). English ASR is included in the training of both tasks to ensure good performance. Table 6.9, shows the Spanish ASR and Spanish to English SLT perform best with the set of similar languages. This is despite the greater number of utterances from the other group of non-related (32752 utterances non-related, 24145 related).

<table>
<thead>
<tr>
<th>ASR model</th>
<th>Es WER</th>
<th>SLT model</th>
<th>Es-En BLEU</th>
<th>Utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>En, Es, It, Pt – ASR</td>
<td>27.5</td>
<td>En – ASR, Es,It,Pt-SLT: En</td>
<td>15.8</td>
<td>24145</td>
</tr>
<tr>
<td>En, Es, Fr, De – ASR</td>
<td>29.3</td>
<td>En – ASR, Es,De,Fr-SLT: En</td>
<td>13.3</td>
<td>32752</td>
</tr>
</tbody>
</table>

Table 6.9: Scores for Es direction with related languages (above) and non-related (below)

## 6.4 Transfer learning evaluation

From the results on joint ASR and SLT training we present our thoughts on the relationship between tasks. To recapitulate the findings, ASR tasks improve the ASR performance and allow SLT to take place effectively. SLT improves SLT performance (when well performing ASR is also present in the training) but does
not improve ASR performance. Within ASR, low-resource benefits greatly from more ASR (either low-resource or high-resource). Low-resource ASR can have good performance in joint training as long as the individual amount of utterances of the language are not too low, as there exists a point at which performance starts to degrade fast as utterances drop. On the other hand, low-resource ASR gives small benefits to high-resource ASR. SLT and low-resource ASR do not mutually improve each others performance. Training multilingual SLT does not improve mutual performance, but training multilingual low-resource ASR does.

We believe improvement in multilingual ASR is related to the phonetics of the model. This has an effect on both SLT and ASR, that is why both tasks benefit. As SLT is a more complex task, it needs a good acoustic model to train properly. However, low-resource ASR can build on the phonetics of poor acoustic models. The improvements in SLT from multilingual SLT probably come from the target side of the task, as better translations are produced. However SLT translations at this level cannot add much information to rigorous ASR evaluated on WER.
6.5 Accented speech in multilingual ASR

This Section analyses the performance of English ASR on non-native speech. The aim is to assess how monolingual English ASR performs and the effects of multilingual ASR models. Non-native speakers usually bring patterns from their mother tongue into their second language which come out as accented speech. Speakers with the same first language usually find it easier to understand each other when they are speaking a foreign language with a marked accent.

The EU parliament has members from many different European countries, and some of them perform their speeches in English as well as in their native language. Each utterance can be assigned a nationality from the speakers to obtain the scores for different accents. The proportion of utterances by nationality in the training and test set is show in Figure 6.1. The numerical breakdown by country using the counts of speakers, speeches and utterances is shown in Table 6.10 for both the train and test English speech data. Each speaker may have various speeches, and each speech will have various utterances.

![Figure 6.1: Counts for each nationality for train (top) and test set (bottom)](image-url)
6.5.1 Challenges and assumptions

Accent is one of the many phonological properties in the speech of a speaker. There exists also pitch, speed, breathiness and other specific enunciation characteristics that make each speaker unique. Beyond the qualities of a speaker, there exists other factors that can affect the ASR performance, such as recording quality. The existence of so many variables makes abundant data necessary to extract strong conclusions. Furthermore, each speaker is enunciating different speeches, which might on their own account be more difficult to the ASR model depending on whether similar tokens were processed during training and the network could learn to generalise them. By grouping speakers we aim to reduce the effect of other aspects and assume the relevant results will reflect the common trait.

6.5.2 Monolingual English ASR performance of native and non-native speech

We first investigate whether accents have an effect on monolingual English ASR performance. Native English speakers are much more numerous than non-native English speakers in the training data, as shown in Figure 6.1. This is to be expected as non-native speakers will do most of their interventions at the EU parliament in their native language. Grouping into native and non-native allows both groups to have a considerable amount of speakers. If accent is a relevant feature of speech, the accent asymmetry in the training set should be reflected in the performance of the model. Accordingly, the monolingual ASR model should perform better on the native English speech in the test set. The model will train mostly on examples of this type of speech and therefore learn its particular afflictions. If this were not the case, other aspects of speech would be masking the effects of accent. To clarify, this effect is not related to the way native English speak the English language, just to their occurrence inside the train data set. If most English speakers in the train set were from Bulgaria, the speeches in English with Bulgarian accent would be expected to get the best scores. Table 6.11 shows there is a considerable difference between the scores of native and non-native speech ASR. The English ASR models performs 4.3 WER points lower on the non-native speech.
6.5.3 Improvement of English ASR of native and non-native speech in multilingual ASR

Next, we assess whether English multilingual ASR can have an impact on accent performance and equalize the results between native and non-native speech. Two English multilingual ASR models are compared, one trained on 3 additional ASRs (Spanish, French and German) and another trained on 5 additional ASRs (Spanish, French, German, Italian and Portuguese). The performance of the models is calculated for each speaker group. Additionally, the difference in performance between the models for each speaker group is shown as well.

Table 6.12 shows that native speakers still obtain considerably better WER scores than non-native speakers for the multilingual ASR models. The difference in improvement between the results for each speaker group suggests a larger improvement for non-native languages by a small margin: 0.6 WER points difference in improvement for the model with 3 auxiliary ASRs and 1.3 WER points difference for 5 ASR.

However, the initial WER for each set of speakers is different. Native speakers (22.8) start at a much lower score than non-native ones (27.1). As WER score gets closer to zero in ASR, models require more data to improve the performance because the relationship between WER score and data is not linear. Therefore, a lower improvement at lower WER scores can be as important as a higher improvement at higher WER scores. Knowing the exact relationship between data and WER, would allow comparing both the native and the non-native improvements for the multilingual ASR model at their respective initial values with the expected improvement, and provide a more mathematical comparison. With the current level of information and the remaining difference in performance between native and non-native performance in the multilingual models, the improvements do not appear remarkable.

Table 6.11: WER breakdown by native or non-native speaker. Next to them statistics on number of speakers, speeches and counts.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Baseline WER</th>
<th>spk</th>
<th>spc</th>
<th>cts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>22.8</td>
<td>8</td>
<td>29</td>
<td>302</td>
</tr>
<tr>
<td>Non Native</td>
<td>27.1</td>
<td>48</td>
<td>98</td>
<td>965</td>
</tr>
</tbody>
</table>
Table 6.12: WER improvement with additional ASR directions.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Baseline</th>
<th>+3 ASR</th>
<th>+5 ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>22.8</td>
<td>20</td>
<td>20.2</td>
</tr>
<tr>
<td>Non Native</td>
<td>27.1</td>
<td>23.7</td>
<td>23.2</td>
</tr>
<tr>
<td>Diff</td>
<td>0</td>
<td>2.8</td>
<td>2.6</td>
</tr>
<tr>
<td>Non-Native</td>
<td>0</td>
<td>3.4</td>
<td>3.9</td>
</tr>
</tbody>
</table>

The idea behind this analysis was that additional ASR languages in the multilingual models would help the non-native European accents more than the native English accents. However, this hypothesis might have been misguided as every language has its own accent. The non-native group includes accents from many nationalities. It is reasonable to assume that English speech with a Spanish accent might benefit more from joint training with Spanish ASR than English speech with an English accent. The English spoken by the Spanish speakers might have traces of Spanish phonetics that the model learns to interpret better thanks to the Spanish ASR. However, there is no reason to assume Swedish accents will benefit more than English accents in the same situation. The transfer that each accent receives from a different language ASR most probably lies in a spectrum which depends on how close the phonetics of the two languages are (and possibly other language factors). Under this view the fact that foreign accent as a group does not improve much more than English accent seems reasonable.

However this explanation does not bar the option that an auxiliary ASR might help its own accented English speech. Furthermore, there is the possibility that the transfer from multilingual ASR does not impact accents in a special manner, and benefits speech according to other criteria, such as phonetical aspects not related to accent.

6.5.4 English ASR results by individual nationalities

Monolingual English ASR scores for each nationality next to the scores of bilingual ASR models for Spanish, French and German, and a multilingual ASR model with the three languages are shown in Table 6.13. The difference in WER scores have the sign changed for clarity. As there are many nationalities present, some groupings include very few utterances. The Multilingual
ASR model trained on English, Spanish, French and German improves on the monolingual ASR for all nationalities except Belgium and Finland.

As mentioned previously, apart from accents, many other possible factors can affect the performance of a given group of sentences, this include the speaker, the recording, and the sentence being recognised. Big groups of utterances use the assumption that other factors average out over many utterances or at act like a small source of noise over the group (accent) signal. Some of the nationalities that obtain the highest or lowest WER results are at the bottom of the table, and have the least number of utterances (Italy, Malta, Finland, Bulgaria, Austria), which supports that volatility occurs with few utterances and speakers.

From the monolingual ASR model it is not possible to determine if some “European accents” have a greater affinity to native English speech, because the non-native speech in the training set might have influenced the performance. The train data includes some non-negligible non-native speech. ~21,400 utterances are in English or Irish while the remaining ~10,000 belong to other European nationalities, as shown in Table 6.10. As an example, Sweden has a high number of speakers and obtains better performance than English for the monolingual ASR model. Plot 6.2 shows the proportion of nationalities in the training data once English and Irish are removed. Sweden is the most numerous nationality in terms of utterances. Therefore it is not possible to state if it is the English utterances from Swedish speakers in the train set or an affinity with the English speakers phonetics what gives Sweden an increase in performance.
Table 6.13: WER by nationality for different additional ASR languages. The signs of the differences in WER are turned (positive indicates a decrease in WER)

<table>
<thead>
<tr>
<th>Nationality</th>
<th>Stats</th>
<th>WER</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cts</td>
<td>spk</td>
<td>Base</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>302</td>
<td>8</td>
<td>22.8</td>
</tr>
<tr>
<td>Romania</td>
<td>155</td>
<td>6</td>
<td>28.8</td>
</tr>
<tr>
<td>Netherlands</td>
<td>74</td>
<td>3</td>
<td>25.8</td>
</tr>
<tr>
<td>Poland</td>
<td>70</td>
<td>3</td>
<td>26.7</td>
</tr>
<tr>
<td>Denmark</td>
<td>65</td>
<td>2</td>
<td>21.5</td>
</tr>
<tr>
<td>Hungary</td>
<td>61</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>Slovakia</td>
<td>57</td>
<td>2</td>
<td>29.6</td>
</tr>
<tr>
<td>Germany</td>
<td>57</td>
<td>4</td>
<td>28.8</td>
</tr>
<tr>
<td>Sweden</td>
<td>54</td>
<td>4</td>
<td>22.5</td>
</tr>
<tr>
<td>Lithuania</td>
<td>53</td>
<td>3</td>
<td>28.8</td>
</tr>
<tr>
<td>Portugal</td>
<td>46</td>
<td>2</td>
<td>30.6</td>
</tr>
<tr>
<td>Greece</td>
<td>44</td>
<td>1</td>
<td>28.7</td>
</tr>
<tr>
<td>Belgium</td>
<td>38</td>
<td>2</td>
<td>30.9</td>
</tr>
<tr>
<td>Czech Republic</td>
<td>36</td>
<td>2</td>
<td>28.6</td>
</tr>
<tr>
<td>Spain</td>
<td>33</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>Austria</td>
<td>30</td>
<td>1</td>
<td>33.7</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>28</td>
<td>1</td>
<td>19.3</td>
</tr>
<tr>
<td>Latvia</td>
<td>24</td>
<td>2</td>
<td>24.1</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>17</td>
<td>1</td>
<td>28.3</td>
</tr>
<tr>
<td>Finland</td>
<td>9</td>
<td>1</td>
<td>29.3</td>
</tr>
<tr>
<td>Malta</td>
<td>7</td>
<td>1</td>
<td>21.8</td>
</tr>
<tr>
<td>Italy</td>
<td>7</td>
<td>1</td>
<td>15.2</td>
</tr>
</tbody>
</table>

6.5.5 Additional accented dataset

Given the lack of data to assess the impact of bilingual ASR models on accented speech, we propose a new accented dataset to work with more data and avoid those pitfalls. This is specially necessary for French speakers, as there were none present in the test set. We create this new test set following the steps laid out in the approach. The results of the ASR are show in 6.14

Table 6.14 shows both the weighted WER and the mean WER for the monolingual English ASR, the bilingual models with Spanish, French, and German and the multilingual ASR model for all three (and English). For the weighted WER, French language seems to help French accented speech the most. Similarly, German language helps German accented speech also the most. However,
Spanish language does not do the same for Spanish accented speech. The Spanish results is very different from the original Europarl test set, where Spanish improved greatly in the Spanish ASR bilingual model. Spanish accented speech is clearly the worst performing for any model, by an 8 point WER difference. When we created the test set, some utterance were allowed to be too long, to avoid breaking up sentences, and we think that affects the results. This might also be pointing to a slight fragility in the ASR system. Examining the mean WERs scores without length weighings, Spanish does give better results. As Spanish only provides 7,000 extra utterances compared to the 12,000 of the other two bilingual models, this could explain why it does not do as well with the longer sentences.
<table>
<thead>
<tr>
<th>Nationality</th>
<th>Baseline</th>
<th>+ Es</th>
<th>+ Fr</th>
<th>+ De</th>
<th>+ All</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weigthed WER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>33</td>
<td>33.8</td>
<td>31</td>
<td>33.9</td>
<td>31.2</td>
<td>19.9</td>
</tr>
<tr>
<td>Germany</td>
<td>33.3</td>
<td>30.9</td>
<td>28.8</td>
<td>28.2</td>
<td>30.1</td>
<td>19.8</td>
</tr>
<tr>
<td>Spain</td>
<td>43.4</td>
<td>42.3</td>
<td>42.6</td>
<td>40.9</td>
<td>38.5</td>
<td>23.3</td>
</tr>
<tr>
<td>Mean WER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>34</td>
<td>34.8</td>
<td>32.8</td>
<td>35.6</td>
<td>31.1</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>32.5</td>
<td>30.3</td>
<td>28.2</td>
<td>26.9</td>
<td>29.6</td>
<td></td>
</tr>
<tr>
<td>Spain</td>
<td>40.8</td>
<td>39.7</td>
<td>42.8</td>
<td>41.9</td>
<td>37.8</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.14: Results artificial dataset by nationality speakers
<table>
<thead>
<tr>
<th></th>
<th>Test</th>
<th></th>
<th></th>
<th>Train</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spk</td>
<td>Spc</td>
<td>Cts</td>
<td>Spk</td>
<td>Spc</td>
<td>Cts</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>8</td>
<td>29</td>
<td>302</td>
<td>79</td>
<td>1659</td>
<td>16847</td>
</tr>
<tr>
<td>Romania</td>
<td>6</td>
<td>16</td>
<td>155</td>
<td>8</td>
<td>86</td>
<td>887</td>
</tr>
<tr>
<td>Netherlands</td>
<td>3</td>
<td>6</td>
<td>74</td>
<td>4</td>
<td>66</td>
<td>796</td>
</tr>
<tr>
<td>Poland</td>
<td>3</td>
<td>7</td>
<td>70</td>
<td>6</td>
<td>30</td>
<td>468</td>
</tr>
<tr>
<td>Denmark</td>
<td>2</td>
<td>6</td>
<td>65</td>
<td>1</td>
<td>8</td>
<td>117</td>
</tr>
<tr>
<td>Hungary</td>
<td>5</td>
<td>6</td>
<td>61</td>
<td>6</td>
<td>44</td>
<td>635</td>
</tr>
<tr>
<td>Slovakia</td>
<td>2</td>
<td>6</td>
<td>57</td>
<td>2</td>
<td>12</td>
<td>200</td>
</tr>
<tr>
<td>Germany</td>
<td>4</td>
<td>6</td>
<td>57</td>
<td>2</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>Sweden</td>
<td>4</td>
<td>5</td>
<td>54</td>
<td>8</td>
<td>74</td>
<td>1205</td>
</tr>
<tr>
<td>Lithuania</td>
<td>3</td>
<td>10</td>
<td>53</td>
<td>2</td>
<td>35</td>
<td>321</td>
</tr>
<tr>
<td>Portugal</td>
<td>2</td>
<td>3</td>
<td>46</td>
<td>2</td>
<td>56</td>
<td>491</td>
</tr>
<tr>
<td>Greece</td>
<td>1</td>
<td>7</td>
<td>44</td>
<td>2</td>
<td>12</td>
<td>235</td>
</tr>
<tr>
<td>Belgium</td>
<td>2</td>
<td>3</td>
<td>38</td>
<td>2</td>
<td>21</td>
<td>372</td>
</tr>
<tr>
<td>Czech Republic</td>
<td>2</td>
<td>2</td>
<td>36</td>
<td>3</td>
<td>29</td>
<td>499</td>
</tr>
<tr>
<td>Spain</td>
<td>1</td>
<td>3</td>
<td>33</td>
<td>2</td>
<td>10</td>
<td>137</td>
</tr>
<tr>
<td>Austria</td>
<td>1</td>
<td>3</td>
<td>30</td>
<td>2</td>
<td>12</td>
<td>313</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>1</td>
<td>3</td>
<td>28</td>
<td>4</td>
<td>34</td>
<td>398</td>
</tr>
<tr>
<td>Latvia</td>
<td>2</td>
<td>2</td>
<td>24</td>
<td>1</td>
<td>16</td>
<td>258</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finland</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>8</td>
<td>128</td>
</tr>
<tr>
<td>Malta</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>36</td>
<td>600</td>
</tr>
<tr>
<td>Italy</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>54</td>
</tr>
<tr>
<td>Ireland</td>
<td>19</td>
<td>515</td>
<td>4614</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyprus</td>
<td>3</td>
<td>57</td>
<td>714</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estonia</td>
<td>4</td>
<td>43</td>
<td>467</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Luxemburg</td>
<td>1</td>
<td>20</td>
<td>368</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slovenia</td>
<td>1</td>
<td>15</td>
<td>249</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bulgaria</td>
<td>1</td>
<td>13</td>
<td>134</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.10: Counts for each nationality for train and test set
Chapter 7

Conclusion

How can we improve ASR in a multilingual setting?

ASR shows the best performance for the multilingual ASR models. The performance of the monolingual ASR baselines is good for English ASR at 30,000 utterances but low (> 50 WER) for the other baselines at 13,000 utterances or less. The low-resource languages obtain good ASR performance for bilingual ASR models trained on the low-resource language and English. Multilingual ASR models improve on these scores. These multilingual ASR models do not need to be trained on a high-resource ASR language to perform well in ASR for the other languages. This could be advantageous to train models for similar groups of low-resource languages. We believe the good performance from multilingual ASR is caused by the different ASR components in training creating a good acoustic model.

How can we improve SLT in a multilingual setting?

SLT performs poorly for the single SLT models. The Multilingual SLT model obtains poor SLT performance as well. Multi task models trained on joint ASR and SLT give much better performance for SLT. The ASR component in the training needs to obtain good WER scores for this, but the relationship to the SLT languages does not matter. This result reinforces the idea that the ASR components in training create a good acoustic model. The SLT task uses the acoustic model to learn to translate well. Models trained on joint ASR and multilingual SLT improve the performance on SLT over models trained with ASR and single SLT. We believe the additional SLT tasks in training are
contributing to the translation performance, teaching the model to perform better translation. This is likely why joint training SLT and ASR does not help ASR performance.

**How can we improve ASR in a multilingual setting?**

The monolingual English ASR model shows there are discrepancies on performance on native and non-native scores. Native score is 4.3 WER higher. Multilingual ASR models improve the performance of English ASR on both groups, but the improvement appears similar taking into account the initial WER values of each group. An additional dataset is proposed for French, Spanish and German accented speech. The performance of bilingual ASR models trained on English and the language of the accent indicate these models perform better for their own accent.
Bibliography


modeling with LSTMs. In 2018 IEEE spoken language technology workshop (SLT) (pp. 174–181).


