Zurich University
of Applied Sciences

School of
Engineering

Bachelor Thesis

ZHAW School of Engineering

CAI, Centre for Artificial Intelligence

Exploring Wav2Vec2 Pre-Training on Swiss German Dialects using Speech Translation and Classification

Authors:
Samuel Stucki
Patrik Randjelovic

Supervisor:
Prof. Dr. Mark Cieliebak

Secondary Supervisor:
Dr. Jan Deriu

June 17, 2022

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

DECLARATION OF ORIGINALITY

Bachelor's Thesis at the School of Engineering

By submitting this Bachelor's thesis, the undersigned student confirms that this thesis is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the Bachelor thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date:	**Name Student:**
Winterthur, 17.06.2022	Samuel Stucki 
Winterthur, 17.06.2022	Patrik Randjelovic 

Abstract

Low-resource languages and dialects, such as Swiss German, require systems that can generalize over several languages to develop state-of-the-art speech translation and recognition applications. This thesis tests the capabilities of the Transformer-based pre-trained cross-lingual Wav2Vec2-XLS-R model on Swiss German corpora. We experiment with both a speech translation system from Swiss German to Standard German and a classification system, assigning dialects to one of four regions, for evaluation. We apply 2100 hours of unlabelled Swiss German data in a pre-training setup to explore the impact this data has on the already pre-trained model. The result of the thesis is a translation system that achieves 18.08% WER and 68.86 BLEU on the "SNF" test corpus and 68.05 BLEU on the "SDS-200" test split. It ranked first in the SwissText "2nd Swiss German Speech to Standard German Text" shared task with 68.1 BLEU on the private evaluation split. The classification task of categorising dialects into four distinct regions achieves a weighted F1-score of 0.49, with the best region reaching 0.68 F1. We showed that providing additional pre-train data at this scale to the XLS-R model is not beneficial for speech translation, but can have a positive impact during classification. By discussing potential approaches for future research we hope to increase the interest on this topic.

Zusammenfassung

Sprachen und Dialekte mit geringen Datenmengen, wie z.B. Schweizerdeutsch, erfordern Systeme, die über zahlreiche Sprachen hinweg verallgemeinern können, um moderne Sprachübersetzungs- und Erkennungsanwendungen zu entwickeln. Diese Arbeit testet die Fähigkeiten des Transformer-basierten, vortrainierten, sprachübergreifenden Wav2Vec2-XLS-R Modells auf schweizerdeutschen Korpora. Wir werten das System anhand einem Sprachübersetzungssystem von Schweizerdeutsch auf Standarddeutsch als auch einem Klassifikationssystem, welches Dialekte vier Regionen zuweist, aus. Wir wenden zusätzlich 2100 Stunden unlabelled schweizerdeutscher Daten in einem pre-training Verfahren auf das Modell an, um die Auswirkungen dieser Daten auf das bereits vortrainierte System zu untersuchen. Das Ergebnis dieser Arbeit ist ein Übersetzungssystem, das 18,08% WER und 68,86 BLEU auf dem "SNF" Testkorpus und 68,05 BLEU auf dem "SDS-200" Test-Split erreicht. Es belegte den ersten Platz im "2nd Swiss German Speech to Standard German Text" SwissText shared task mit 68,1 BLEU auf dem privaten Evaluationssplit. Im Klassifikationsexperiment, welches Schweizer Dialekte in vier verschiedene Regionen kategorisiert hat, wird ein gewichteter F1-Score von 0,49 erreicht, wobei die Ostschweizer Region den besten F1 mit 0,68 erzielt. Wir haben gezeigt, dass die Verwendung zusätzlicher pre-train Daten in dieser Grössenordnung beim XLS-R-Modell für die Sprachübersetzung nicht von Vorteil ist, sich aber bei der Klassifizierung positiv auswirken kann. Mithilfe einer Diskussion für zukünftige Forschungsansätze hoffen wir, dass das Interesse für dieses Themengebiet steigt.

Preface

This work has given us the opportunity to delve deep into ASR-related topics, which would otherwise have not been possible. Having had the opportunity to work and exchange fascinating ideas with multiple experts in the field made this thesis even more interesting. The experience has proven itself to be immensely valuable for our personal and professional growth. This thesis is directed at prospects that would like to follow the advancements made for Swiss German STT translation and speech classification systems.

We would like to thank our supervisors Prof. Dr. Mark Cieliebak and Dr. Jan Deriu for their invaluable input and support during the duration of the thesis. We also want to thank the ZHAW CAI and InIT department for providing the infrastructure without which we could not have tested our hypothesis. Special thanks also go to Zhviar Sourati Hassanzadeh for performing the webcrawling of the SRF data and to Katsiaryna Mlynchyk for the access to their Voice Activity Detection system. Finally, we would also like to thank the organisers of the SwissText conference and the shared task organisers for their tireless efforts to advance the Swiss NLP landscape.

Contents

Chapter 1

Introduction

Recent years have seen exponential growth in using neural networks to solve problems involving speech classification and recognition, along with transcription and translation. This is due to its ability to work with incomplete knowledge on the domain of the task [1] and the reduced complexity of implementation provided through a multitude of different libraries. These problems belong to the Natural Language Processing (NLP) subbranch in Artificial Intelligence (AI), which in the last decade has see a shift from the traditional statistical approach to the aforementioned neural network approach involving Deep Learning (DL).

Switzerland is a federation comprising 26 cantons with four official languages: German, French, Italian, and Romansh. These languages are spoken in many dialects, with Swiss German making up the largest language group, being adopted by 21 of the 26 cantons and spoken by 62.3% of the population [2]. Automatic Speech Recognition (ASR) has not been as effective for Swiss German as for other languages, based on the need for large quantities of training data in neural networks. Compared to Standard German, English, or Mandarin, Swiss German belongs to so-called low-resource languages, which often lack the required data quantities for training neural models.

The barrier for low-resource languages has been largely lifted with the release of Transformers-based pre-trained models like BERT [3] and Wav2Vec [4]. The efforts of SwissNLP and other Swiss institutions to create publicly accessible corpora like the SDS-200 [5] have now given the Swiss public the tools needed to implement Swiss German NLP systems like Speech-to-Text (STT) translation or recognition.

In this thesis, we seek to implement a system that can translate Swiss German dialects into Standard German and classify them into four regions. By utilising the pre-trained cross-lingual Wav2Vec2-XLS-R model, we want to research the impact of additional unlabelled Swiss German pre-training data on the model to improve previous results achieved in the area. We compare our results with previous work by using downstream fine-tuning on STT translation and classification. By entering our models to the SwissText Swiss German Speech to Standard German Text shared task [6], we want to test their capabilities in a competitive setting.

1.1 Literature review

Natural Language Processing is a subbranch of Artificial Intelligence with roots in linguistics that concerns itself with giving computers the ability to understand text and speech, including contextual nuances of the languages within them, to help analyze and interpret human languages. Automatic Speech Recognition is one of the various subtasks of NLP and aims at building systems that can automatically recognize and transcribe speech into text. Within this discipline is the underlying subject of STT translation and Dialect Identification (DID). Many traditional approaches have recently been replaced by neural-network-based pre-trained Transformer models [7], such as the 2019 introduced text-based BERT model [3]. Several new Transformer-based architectures have since emerged and are currently state-of-the-art (SOTA) in the NLP world. References for different applications in the tasks of pre-training, STT translation, and dialect classification, which are applied in this dissertation, will be provided.

In December of 2020 Abdul-Mageed et al. released two Arabic pre-trained BERT-based models called ARBERT (standard) and MARBERT (for dialects). The ARBERT model is trained on Modern Standard Arabic (MSA) sources. Alongside the ARBERT model, the so-called MARBERT model was released as well, since the Arabic language contains a large number of diverse dialects. After fine-tuning the models on multiple tasks such as sentiment analysis, topic classification, and DID they outperformed previous SOTA F1 scores by sizeable margins with the best score reaching 90.89% in the QADI country-level dialect corpus. They were thus able to display the positive impact task-specific pre-training can have on a language or dialect. [8][9]

Speech translation using Wav2Vec has been extensively tested by Wu et al. and showed that the self-supervised approach of the model is capable of improving translation performance. They used the model on a setup for both English-to-X and X-to-English translations. On the translations from English to a different language the model achieved a BLEU of 29.8 for French and 28.2 for Romanian. On the task of translating other languages into English, the model achieved the best performance on French-to-English with a BLEU score of 23.00. [10]

Publications on Swiss German speech translation systems have also increased over of the last years. Garner et al. [11] used Hidden Markov models to transcribe the Valais dialect into Standard German achieving a WER of 19.4% in 2014. In 2020 the Swiss-Text shared task "GermEval 2020 Task 4: Low-Resource Speech-to-Text" [12] provided participants with 74 hours of speech originating predominantly from the Bernese parliament to create a speech translation system from Swiss German to Standard German. The best performing team reached a WER of 40.29% using the Jasper CNN acoustic model [13]. In the first SwissText "Swiss German speech to Standard German text" shared task was held in 2021 [14] to create a model that can transcribe Swiss German to Standard German. The metric used for evaluating the submissions was BLEU. The best and only team to beat the baseline model was

Microsoft, which used a hybrid system incorporating a translation lexicon, a first pass language model with Swiss German data, an acoustic model based on transfer learned Standard German data, and a second pass neural language model for pass rescoring. They achieved a BLEU score of 46.04 with the baseline having a BLEU of 41.0. ZHAW also submitted a solution based on ensembling three different approaches, with one of these applying the XLSR-53 model. They achieved a BLEU score of 39.4 and reached second place when discounting the baseline model. Subsequently, on the 4th of October 2021, Microsoft announced a general availability release of Swiss German speech recognition and transcription on their Azure platform [15].

Dialect classification of Finnish dialects was performed by Hämäläinen et al. from the University of Helsinki. They compared two separate approaches to for classification. The first one was text only using a bi-directional long short-term memory (LSTM) based model. The second and much better performing approach was a combination of text and audio using a siamese neural network architecture that combined both BERT [3] and the Wav2Vec 2.0 XLSR-53 [16][17] architecture. They used the FinBERT model released in [18] with a Finnish fine-tuned XLSR-53 model. By implementing a fixed input length and a global average pooling for BERT using an adaptive average pooling for the Wav2Vec part of the model they ensured that each side produced an equal size of features. The results showed that the combinatory model had consistently higher scores than the text-only based model with multiple dialects having an F1-score of more than 0.90 while the best text-only model score was 0.75. It indicated that the audio contained features that were not sufficiently covered by the transcriptions in the dataset used by the team. Additionally, an observation was made where dialects with low sample sizes still had high F1-scores suggesting that large amounts of data for any single dialect do not necessarily have to result in high scores. [19]

For Swiss German, the VarDial 2019 shared task "Third German Dialect Identification (GDI)" outlined a setup in which texts originating from the regions of BE, BS, LU, and ZH had to be classified. The best team reached a macro F1 of 75.93% with the ZHAW TwistBytes team of Benites et al. [20] reaching 74.55% macro F1. [21]

In 2021 we performed four experiments on Swiss German dialects using the Wav2Vec2-XLSR-53 model in the scope of a project thesis. Using an smaller unreleased version SDS-200 [5] corpus we tested a Swiss German fine-tuned version of XLSR-53 [22] to differentiate the various dialects based on different grades of granularity. The best performing experiment was categorising the dialects into four distinct regions based on their linguistic similarity and geographic proximity. The model achieved a macro F1 of 45.96% and a weighted F1 of 0.5.

1.2 Outline

This work begins with an introduction (Chapter 1) of the various questions and topics covered and follows the literature review in order to be oriented towards other works and results that are to some extent related to the thesis. In the Foundations Chapter 2, the concepts and instruments used are described with sufficient detail so that the experiments can be comprehensively understood. In order to provide an insight to the data used, Data Collection Chapter 3 contain the necessary information on these. SwissText Shared Task Chapter 4 deals with the matters we had to do for the SwissText conference. The experiments we did are described in the Experimental Setup Chapter 5. This contains the information needed to trace how the results were obtained, which are then presented in the Result Chapter 6. Finally we find the Discussion and Outlook Chapter 7 where the results obtained as a whole, both positive and negative, are interpreted and some possibilities for future work are discussed.

Chapter 2

Foundations

2.1 Speech Processing

Allowing users to interact with machines and devices using their natural verbal language has been a topic of research for a long time. Since the invention of the famous Bell telephone in 1952 [23] numerous advances have been achieved. The progress has been relatively fast from the hidden Markov models to the gaussian mixture models and in the last decade the different neural network architectures. Two current examples of speech processing applications are Apple's Siri, Microsofts's Cortana, and Amazon's Alexa which can process and respond to natural language in a precise manner [24].

2.2 Speech Translation

Speech translation in the context of NLP is a process in which a spoken language is translated automatically from its original language to a target language. The translation can be in form of Speech-to-Text (STT) or Speech-to-Speech (STS). Concerning the globalisation of our world the task has become vital to form a communication bridge for people all over the world [25]. The task has multiple problems which have to be solved concerning the different lengths of the translation sentences, the different vocabulary used, separate alphabets, and differences in grammar. A few of these issues have effectively been solved by the community since the emergence of the field. However, performance improvements are still possible and research in the field thus continues to be strong.

2.3 Dialect Identification

Dialect identification is a subtask of ASR which concerns itself with recognising and classifying dialects. Different from language identification, DID thus requires systems to differentiate parts of the same language family by trying to learn the various uses of grammar and vocabulary. The most common type of dialect is the geographic dialect, characterised by regional boundaries in which they are spoken. These borders are not static and dialects in proximity deviate barely from each

other, while differences are more pronounced in regions that are further away. [26] Swiss German is a collection of such dialects. A factor that makes DID a lot more complex is the oftentimes limited amount of labelled text and audio data. [27] As there are no standardisations in these dialects, be that grammar or pronunciation, they also are more susceptible to change over time. Based on these factors, a need arose for an architecture that can sufficiently identify dialects while only having access to a low amount of resource data.

2.4 Transformers

Transformers are a new SOTA encoder-decoder model developed by Vaswani et al. in 2017, designed to handle sequential data such as natural language, which solely uses attention mechanisms by removing the recurrent neural network (RNN) part of the architecture that most systems used until then. RNNs are sequential by design, which prohibits them from being used in a parallel architecture. This has negative implications towards system performance, most notably when working with longer sequence lengths, which impacts memory usage. Tests not only demonstrated significant improvements in performance but also evaluation scores in translation tasks, thus suggesting that Transformers should be able to be used on other tasks as well. They confirmed the thesis with the release of BERT in 2019 [3], which improved evaluation scores on eleven different NLP tasks. As a result, Transformers have become the de facto standard encoder-decoder architecture in NLP. An overview of the most important aspects of Transformers is given to allow for a better understanding of the concepts discussed in this thesis. [7]

2.4.1 Encoder and Decoder

At the base of Transformers lays the encoder-decoder architecture as visualised in Figure 2.1. Based on [7] and [28] an explanation is given on the inner workings of the technology to enable a better understanding. The architecture is built on the sequence-to-sequence model which aims at finding a mapping f based on a sequence of arbitrary length $X_{1:n}$ to a different arbitrary length sequence $Y_{1:m}$, thus forming the relation of $f\colon X_{1:n} \mapsto Y_{1:m}$. In the case of Transformers, both the encoder and decoder are operating as a stack, with each of them having 6 identical layers. Applicability of real-world problems for this kind of model is any type of task generating text like translations from one language to another.

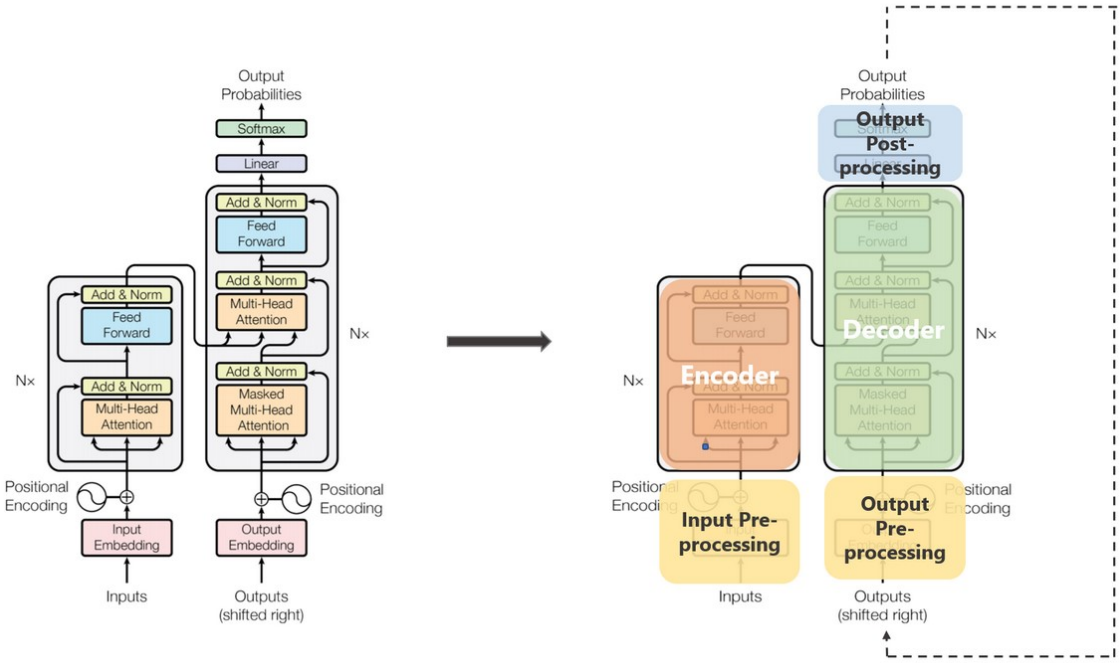


Figure 2.1: Structure of encoder-decoder in the Transformers architecture, figure taken from [7] [29]

Since a sentence depends upon the position of its words to convey the meaning, the same has to be done for the model. In a pre-processing step in both the encoder and decoder, the input is split up into tokens with their respective positional encoding to ensure that the model has access to their relative position. The encoder then encodes the input sequence $X_{1:n}$ to a sequence of hidden states forming $\overline{X}_{1:n}$ of dimension $d_{model} = 512$. Using these encoded hidden states the decoder then models the conditional probability distribution of the target sequence $Y_{1:m}$. Factoring the distribution to a product of the encoded hidden states $\overline{X}_{1:n}$, the distribution of target vector y_i, and all previous target vectors $Y_{0:i-1}$ the decoder can then map these elements to a logit vector l_i. A Softmax function is then applied to the l_i vector, which returns the conditional distribution for y_i. The most important difference to the RNN-based architectures is that this operation explicitly considers all previous target vectors $Y_{0:i-1}$, which was not feasible in the sequential design of the RNNs that only implemented it implicitly. Two special vectors, the end-of-sentence EOS and begin-of-sentence BOS vector, are added to the operation as well. While the EOS vector is applied to both encoder and decoder as the last vector x_n and y_m, respectively, the BOS is only used in the decoder at the 0th position y_0. The operation is visualised with an example for better understanding in Figure 2.2. [28]

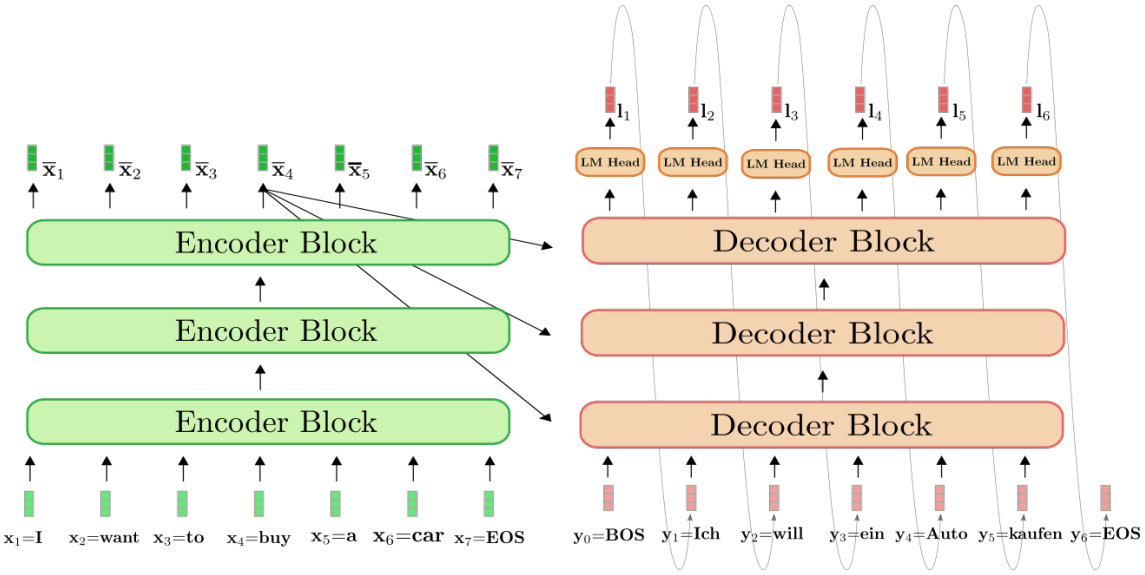


Figure 2.2: Visualization of auto-regressive generation in Transformer-based encoder-decoder model, figure taken from [28]

Each individual hidden state $\overline{X}_{1:n}$ such as $\overline{x}_3$ is not simply dependent on the input counterpart x_3 "to" but on the complete input "I", "want", ..., "car", including x_7 "EOS". This batch of input encodings is then combined with y_0 "BOS" to create l_1 that represents the conditional distribution for y_1. The target vector y_1 is then sampled from l_1 and fed back into the decoder including all previously used target vectors $Y_{0:i-1}$, which in this case is y_0, to create the conditional distribution of the next target vector y_2. This operation continues in an auto-regressive fashion. Important to note is that the encoder operation is only performed once, as portrayed on the right side in Figure 2.1. Afterwards, the decoder is handling the operations on its own in a loop by reusing the calculated input encodings $\overline{X}_{1:n}$.

2.4.2 Attention

Attention was first introduced by Bahdanau et al. in 2014 [30] to provide a neural architecture that can dynamically highlight relevant features in both raw inputs and higher-level representations of said input [30][31]. Transformers extensively utilise attention, specifically self-attention, as explained in section 2.4 by using the mechanism inside its encoder-decoder architecture [7].

Self-Attention

To understand the operations performed inside each encoder and decoder sub-layer, as seen in Figure 2.2, the self-attention mechanism has to be explained by taking examples from [29], [32] and [33]. In essence, self-attention allows for n inputs to interact with each other (the "self" part of the term) and then calculates to which of these it should pay more attention to (the "attention" part). After having finished calculating, it aggregates the interactions and attention scores to n outputs. The

operation performed in this step is termed as "Scaled Dot-Product Attention" by Vaswani et al. [7], which is illustrated in Figure 2.3.

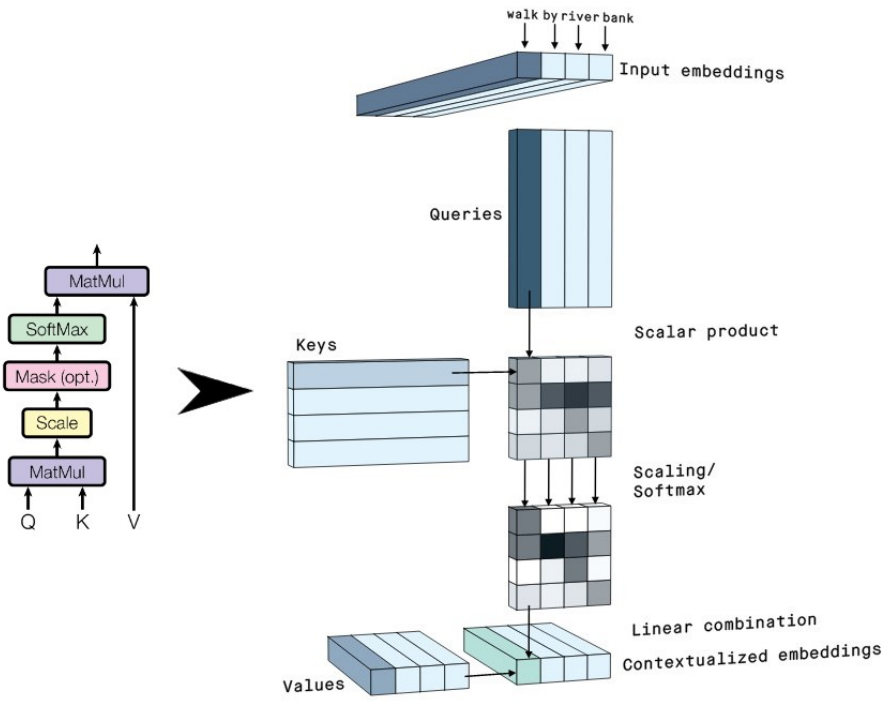


Figure 2.3: Scaled Dot-Product Attention, original figure on the left taken from [7] and the more precise version on the right taken from [32]

Calculating the scalar product directly on the input embedding would lead to an issue where similar tokens receive a higher value and dissimilar tokens a lower value. Relationships between tokens that would otherwise be important in a linguistic setting like that of a subject and a verb or a preposition and a complement are ignored in this way [32]. Attention thus introduces three projection vectors to combat this issue: keys K, queries Q and values V. These are calculated by multiplying the input vector x_i with a corresponding weight matrix W_k, W_q or W_v, which is randomly initialised at first and then taken as a learning parameter by the model. Each vector has a different role: Keys are vectors used to calculate attention against and could be seen as an indexing mechanism for values V, similar to a database. Queries can be thought of as the currently processing token for which the attention is calculated for. Values are used to apply attention, where each value v_i and its corresponding key k_i can provide two different interpretations of the same entity [31]. An important difference is the source from these vectors are created from. In the encoder K, Q and V are all from the same document while in the decoder the Q is from the target document while K and V are both from the source document. [7][29][33]

As seen in Figure 2.3, the first operation is to take a newly calculated query Q and, by performing a dot product with all keys K, find the most similar key. The higher the resulting score for each query-key pair, the closer their relationship to each other is. After this, a scaling operation is performed on the result by dividing it

with the square root of the dimensions of the key vector d_k to counteract a problem where large values in the dot product result in pushing the subsequent Softmax operation to regions where only extremely small gradients exist [7]. To pronounce higher and lower scores, the previously mentioned Softmax operation is applied by driving them more towards 1 and 0, respectively. At last, the Softmax distribution is multiplied with values V where values multiplied by scores closer to 1 will receive more attention than those closer to 0. The resulting equation 2.1 is shown below.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Multi-Head Attention

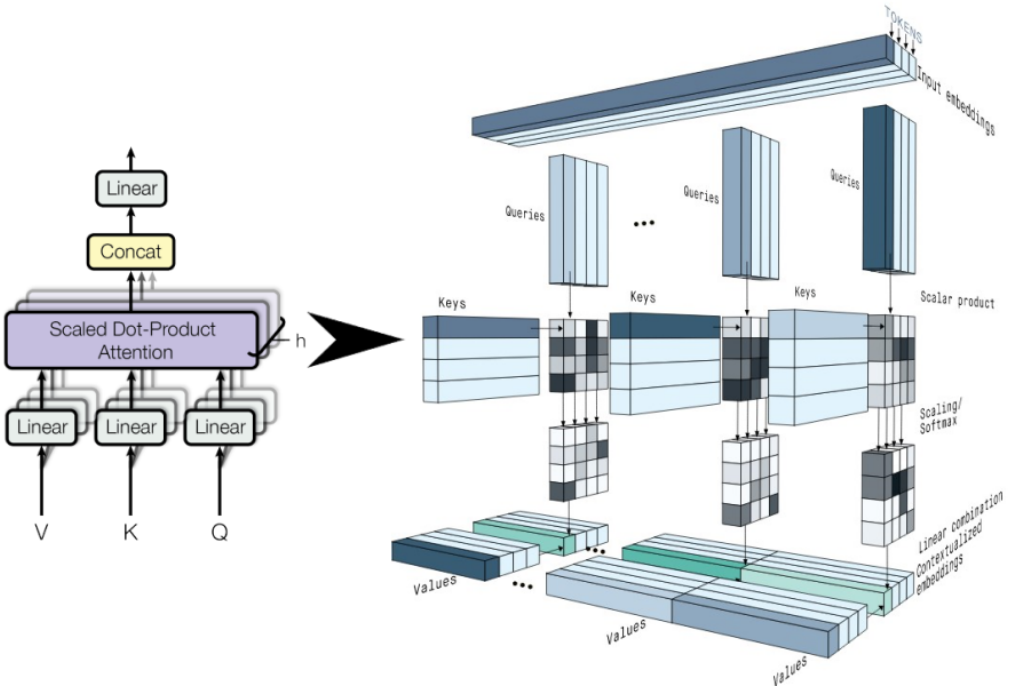


Figure 2.4: Architecture of Multi-head attention, original figure on the left taken from [7] and the more precise version on the right taken from [32]

The creators of Transformers found it beneficial to apply attention in a parallel multi-layer setup by stacking $h = 8$ self-attention layers together where each layer would concentrate on a different set of K, Q and V, which allows the model to explore different relations for the same tokens. First, the three vectors and their corresponding weight matrices W_k, W_q or W_v are reduced in size. All h sets of K, Q and V are termed as "attention head" and instead of averaging the generated outputs like a single attention head would, a concatenation operation is performed to form one large contextualised embedding [29]. This embedding is then multiplied by a last weight matrix W^O. By reducing the dimensions of the 8 attention heads,

the team could bring the computational cost to a similar level of a single attention layer with the complete dimensionality of the vectors and weight matrices. The operation is formalised in equation 2.2 and visualised in Figure 2.4. [7]

$$\begin{aligned} MultiHead(Q, K, V) &= \text{Concat}(head_1, ..., head_h)W^O \\ \text{where } head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

Types of Multi-Head Attention used

Transformer employs three different implementations of multi-head attention, all of which are depicted in Figure 2.5. The first type is the input self attention used in the encoder where the hidden encodings are created as explained in section 2.4.1.

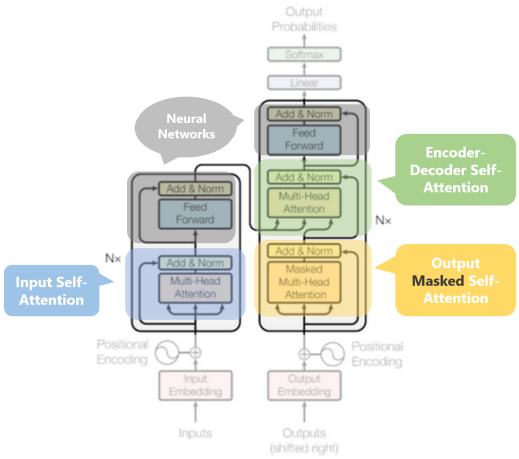


Figure 2.5: Types of multi-head attention inside transformers, figure taken from [29]. Marked in blue is the encoder input self-attention, in yellow the decoder output masked self-attention, and in green the encoder-decoder self-attention

The second type is the decoder output masked self-attention layer, which is similar to the first type but differs in that it performs a masking operation on the input. A sequence mask is applied because of the parallel architecture of Transformers, which does not have a built-in mechanism to prohibit comparisons from being made after a time step t. This would allow the model to generate predictions with future information. As discussed in section 2.4.1, a decoded output y_i should only depend on itself and the previous outputs y_0, ..., y_{i-1} and predict the subsequent vector y_{i+1}. Having information about y_{i+1} ahead of the prediction would seriously impair the training and thus needs to be restricted. The resulting masking matrix can differ based on the chosen visualisation, but in this example, an upper triangular matrix is introduced, represented in Figure 2.6 where the upper violet half is initially filled by $-\infty$ and then transformed to zeros by the Softmax operation. An additional visualisation with values is provided in Figure 2.7. [34]

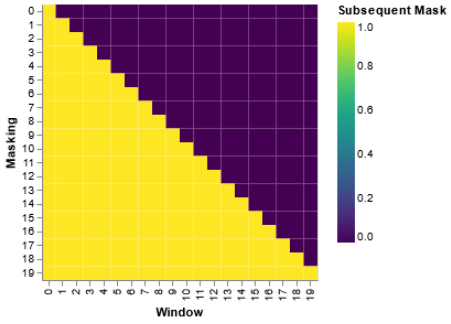
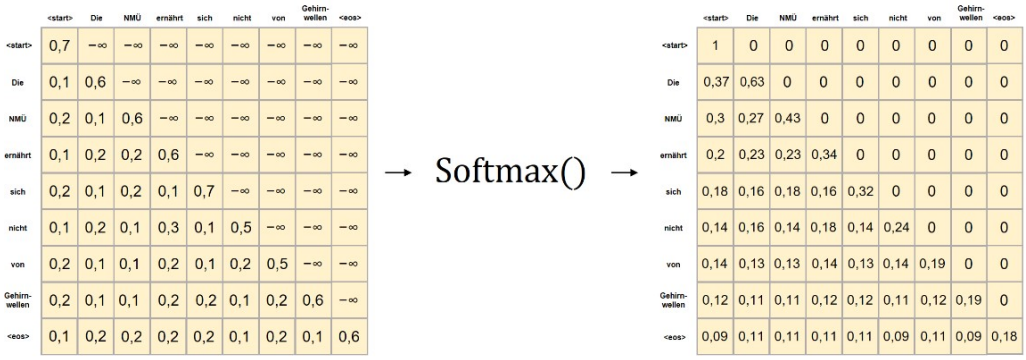


Figure 2.6: Upper triangular matrix, figure taken from [35]

Masking all subsequent words with zeros removes the possibility of y_1 accessing information about y_2, ..., y_m as an example. This mask was termed as "Look-Ahead mask" by the the authors [7].



	<start>	Die	NMÜ	ernährt	sich	nicht	von	Gehirn-wellen	<eos>
<start>	0,7	−∞	−∞	−∞	−∞	−∞	−∞	−∞	−∞
Die	0,1	0,6	−∞	−∞	−∞	−∞	−∞	−∞	−∞
NMÜ	0,2	0,1	0,6	−∞	−∞	−∞	−∞	−∞	−∞
ernährt	0,1	0,2	0,2	0,6	−∞	−∞	−∞	−∞	−∞
sich	0,2	0,1	0,2	0,1	0,7	−∞	−∞	−∞	−∞
nicht	0,1	0,2	0,1	0,3	0,1	0,5	−∞	−∞	−∞
von	0,2	0,1	0,1	0,2	0,1	0,2	0,5	−∞	−∞
Gehirn-wellen	0,2	0,1	0,1	0,2	0,2	0,1	0,2	0,6	−∞
<eos>	0,1	0,2	0,2	0,2	0,2	0,1	0,2	0,1	0,6

	<start>	Die	NMÜ	ernährt	sich	nicht	von	Gehirn-wellen	<eos>
<start>	1	0	0	0	0	0	0	0	0
Die	0,37	0,63	0	0	0	0	0	0	0
NMÜ	0,3	0,27	0,43	0	0	0	0	0	0
ernährt	0,2	0,23	0,23	0,34	0	0	0	0	0
sich	0,18	0,16	0,18	0,16	0,32	0	0	0	0
nicht	0,14	0,16	0,14	0,18	0,14	0,24	0	0	0
von	0,14	0,13	0,13	0,14	0,13	0,14	0,19	0	0
Gehirn-wellen	0,12	0,11	0,11	0,12	0,12	0,11	0,12	0,19	0
<eos>	0,09	0,11	0,11	0,11	0,11	0,09	0,11	0,09	0,18

Figure 2.7: Application of Look-Ahead mask on a word sequence, figure taken from [34]

The encoder-decoder attention layer in Figure 2.5 is the last type. It queries both the output of the masked self-attention layer and the contextualised output of the encoder to formulate a calculation based on both the previously generated target sequences as well as the initial input sequence. Special in this layer is that the query vector is taken from the masked self-attention layer and the key and value vectors from the output of the encoder. This operation has been described in section 2.4.1. [34]

2.5 Wav2Vec

Baevski et al. [4] released Wav2Vec originally in 2019 to reduce the dependency on large amounts of transcribed audio data for training to achieve SOTA performance. It is uniquely difficult to obtain transcribed audio data compared to other types of data because of the presence of time [36]. Labelled audio data needs to be either recorded for a specific sentence or cut into pieces from a larger audio file like a podcast to which afterwards a sentence that matches the spoken speech has to be assigned. Creating corpora for uncommon languages and dialects such as Swiss

German is often not feasible because of both low interest in the corporate world and missing available resources to begin with. Wav2Vec and other similar models like BERT [3] achieve their goals by performing extensive unsupervised pre-training on thousands of hours of unlabelled data. During a second fine-tuning step labelled data is given to the model which then uses the information learned during pre-training to adapt itself and thus reducing the amount of data needed compared to a traditional approach. [4]

The model is a convolutional neural network (CNN) largely based on both the 2017 released Transformer architecture [7], of which an overview can be found in section 2.4, and the 2019 released BERT model [3].

Multiple iterations have since emerged, of which we will only discuss variants of the 2020 released Wav2Vec 2.0 model [16]. Compared to the original model, this and all subsequent versions switched from the unsupervised approach to self-supervision, with which significant improvements in accuracy were achieved. However, during the evaluation, the research team observed that the monolingual nature of the base model was only beneficial for English data and not for other languages. Thus, the cross-lingual Wav2Vec2-XLSR-53 model was created which can generalise across multiple languages and contains about 50k hours of data from 53 different languages in its pre-training step [17]. In 2021, the latest version of Wav2Vec2, the Wav2Vec2-XLS-R model, has been released which seriously improved upon the XLSR-53 model by using 436k hours of training data from 128 different languages [37].

This thesis is based on the XLS-R model and will thus be the primary focus going forward when explaining the inner workings of Wav2Vec2.

2.5.1 Pre-Training

Pre-training is used for so-called transfer learning, which refers to the attempt at training a model with a certain goal to generate parameters that can be used downstream in a different task. The concept is inspired by the human ability to transfer already learned "old" knowledge and apply it in a new but similar setting, thus not having to learn everything from the start again. Transformer apply both aforementioned steps, with the "old" knowledge being learned during pre-training and transferring said knowledge during the fine-tuning phase for a new task. Self-supervision during this pre-training phase enabled Wav2Vec to use unlabelled data and can thus utilise larger amounts of resources that are also more readily available. [16]

2.5.2 Latent Speech Representation

Phonemes are a linguistic concept that represent the smallest unit in speech with which differentiations between similar sounding words or word elements can be made. They are divided into vowels and consonants, of which all languages and dialects have different sets of [38]. This changes, however, when analysing recorded speech waveforms which can exhibit a variance based on emotional state, individual speak-

ing style, linguistic content and more [39]. By trying to extract the smallest distinct elements, the so-called latent speech representation can be captured. Until recently, hand-crafted solutions were the norm, which were expensive to implement in both time and resources. This changed with the advent of Transformer based models where the speech representations could be learned during pre-training using the large amounts of unlabelled data.

2.5.3 Quantization

Values in a continuous space, such as the latent speech representations, have to be brought back into a finite set of values in the discrete space. Wav2Vec2 uses quantization to perform this operation and provides G codebooks with V entries for this. For every latent speech representation z_t, the model takes the best matching entry of each codebook and concatenates them into a vector e_t that is then processed into a quantized representation q_t by a linear transformation to give the model a target for learning by comparison. [16][40]. Figure 2.8 illustrates this operation.

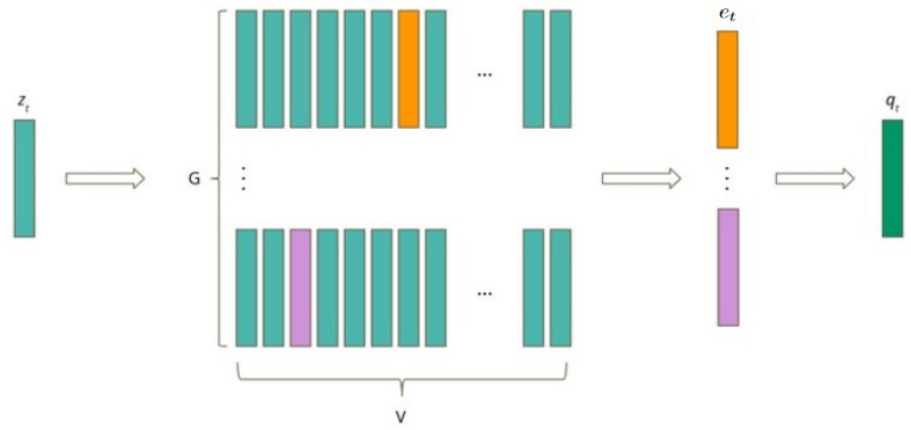


Figure 2.8: Quantization process, figure by Łukasz Sus taken from [40]

To allow a certain freedom during the early stages of training, the Gumbel Softmax randomisation algorithm is introduced to this operation. By allowing randomisation, the model can consider combinations of different entries V that could be beneficial for training, which it otherwise would have dismissed. So-called temperature is used as well to reduce the impact of the randomisation over time by lowering it from the initial value of 2 to 0.5. [16][40]

2.5.4 Masking

Before passing the latent speech representations to the Transformer for further processing, a mask is applied to the input. This mask uses a probability $p = 0.065$ of all representations, also called time steps, to be used as starting indices for subsequent time steps $M = 10$ to be masked as well. Masked inputs will then have to be filled back in by the Transformer architecture and compared to the quantized representations. This step is crucial for the model to learn by comparison. Figure

2.9 depicts an example where index 8 and 12 have been selected by probability p as starting indices, with each then masking the M subsequent time steps. This results 14 masked time steps, as some of them overlap with each other. [16][40]

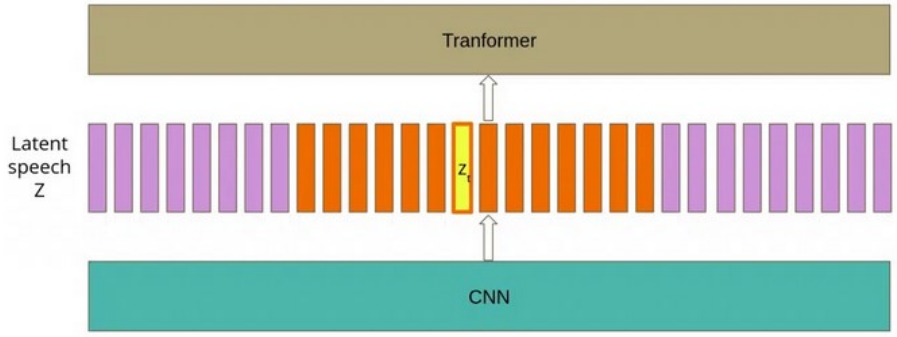


Figure 2.9: Masking process of two indices and the subsequent 10 times steps, figure by Łukasz Sus taken from [40]

2.5.5 Contrastive learning

Contrastive learning is defined in [41] as a concept that aims to learn by comparing among different samples and grouping them into either a "similar" or "dissimilar" group. These two clusters are situated far away from each other in the embedding space to ensure that during contrasting of positive pair samples only the positive representations of the "similar" cluster are pulled together and vice versa for negative pair samples and negative representations of the "dissimilar" cluster. Wav2Vec2 employs this learning mechanism during pre-training to guess the correct quantized representation q_t using the Transformer generated contextualised representation c_t. A benefit of using contrastive methods is that the model architecture does not have to be modified for fine-tuning compared to other self-supervised learning architectures such as [42].

2.5.6 Connectionist Temporal Classification (CTC)

Before a model can make transcribed predictions, the audio representations have to be classified into a sequence of output letters [43]. Early audio models required external Language Models and a dictionary to transform audio frames into a valid transcription. With CTC, developed by Graves et al. [44], an algorithm was created that can automatically learn the alignments of speech and their transcriptions by implementing a loss function that, for a given speech input X, tries to maximize the probability that the produced text output Y is correct. By learning this alignment autonomously the issue with different sizes of input and output is solved as well. CTC can be applied to encoder-decoder architectures, and both encoders and decoders do not have restrictions concerning their implementation. [45][43]

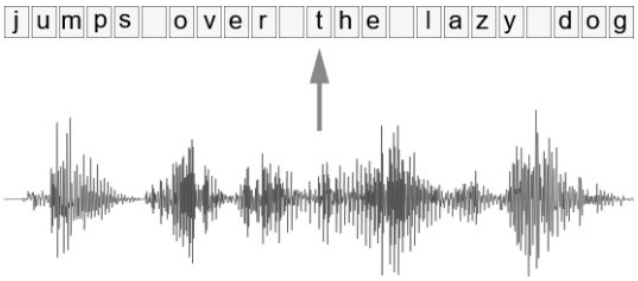


Figure 2.10: Alignment of speech to a transcript, figure taken from [45]

A shortcoming of CTC is that it is conditionally independent, meaning that it assumes that an output is independent of other outputs from the same input. This assumption is erroneous and will be explained with an example from [45]. If an audio sample contains the sequence "triple A" then the transcript could either be "triple A" or "AAA". Should a model predict the first character to be "A" then the first option is not valid anymore. CTC does not take this into account and can thus lead to reductions in accuracy during training. However, if needed, this can be fixed by using an external Language Model on the output of a CTC-based model to boost the performance. The problem is illustrated in Figure 2.11.

Figure 2.11: Issue of conditional independence, figure taken from [45]

Wav2Vec2 leverages the CTC algorithm for fine-tuning and uses the contextualised audio classifications originating from the Transformer layers to remove the dependency on an external Language Model to achieve acceptable transcriptions. [43] The authors of Wav2Vec draw attention to the issue with CTC's conditional independence in their paper [16] by actively promoting the use of LMs to increase the accuracy of the model.

2.5.7 Wav2Vec 2.0 Model Architecture

All aforementioned concepts are combined into the Wav2Vec 2.0 architecture, which is visualised in Figure 2.12. During pre-training, the raw waveform X is fed into the multi-layer CNN feature encoder which yields the latent speech representations Z, forming the relationship of $f\colon X \mapsto Z$. By first masking (see section 2.5.4) and then passing these representations to the Transformer architecture, the contextualised representations C are built, forming $g\colon Z \mapsto C$. As discussed in section 2.5.3, these latent speech representations are discretized with a quantization module to form Q as comparison targets for the model by a relation of $Z \mapsto Q$. [16]

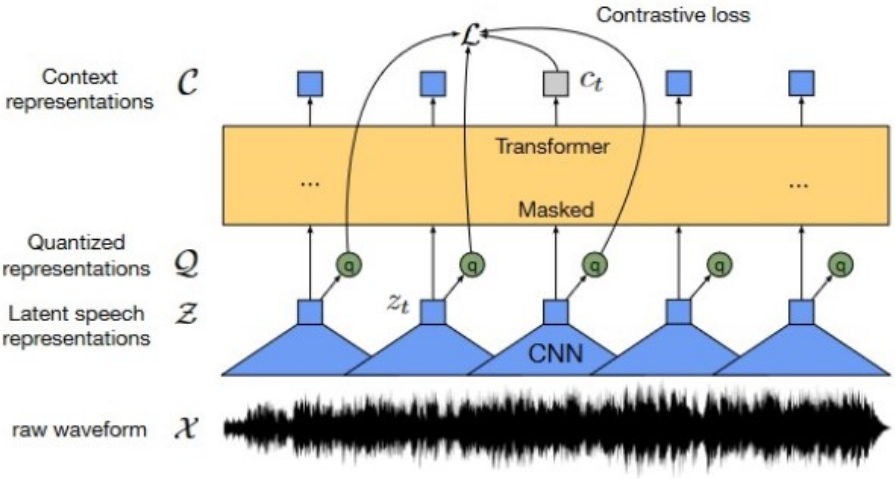


Figure 2.12: Architecture of Wav2Vec 2.0, figure taken from [16]

Each contextualised representation c_t is then compared to a quantized representations q_t using the contrastive loss, formalised in equation 2.3, with which it tries to optimise the Transformer. κ is a temperature which is constant during training and $sim(a, b)$ denotes the cosine similarity. [16]

$$L_m = -log \frac{\exp(sim(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q})/\kappa)} \tag{2.3}$$

Contrastive learning depends upon a varied use of quantized codebooks G with entries V. As a result the diversity loss, formalised in equation 2.4, was introduced which aims at using entropy and maximising it over an averaged Softmax distribution l for all entries in each codebook $\bar{p}_g$ so that the model takes full advantage of the provided code words. [16]

$$L_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} log(\bar{p}_{g,v}) \tag{2.4}$$

The training goal of Wav2Vec2 is thus a sum of both the contrastive loss function L_m and the diversity loss function L_d, forming equation 2.5. [16]

$$L = L_m + L_d \tag{2.5}$$

2.5.8 Wav2Vec2 XLS-R Model Architecture

Wav2Vec2-XLS-R is a large-scale, cross-lingual, speech-representation model proposed by Babu et al. [37] and is based on the Wav2Vec 2.0 architecture by Baevski

et al. [16]. Taking inspiration from Wav2Vec2-XLSR-53 the team aimed at building a more diverse model by using a larger amount of unlabelled training data that includes even more languages. They achieved this by applying 436k hours of data from multiple sources, with the most significant one being the newly released VoxPopuli corpus [46]. Three base versions have been released by the authors based on the number of parameters available during training. From smallest to largest starting with the 300 Million, then the 1 Billion and lastly the 2 Billion model. [37]

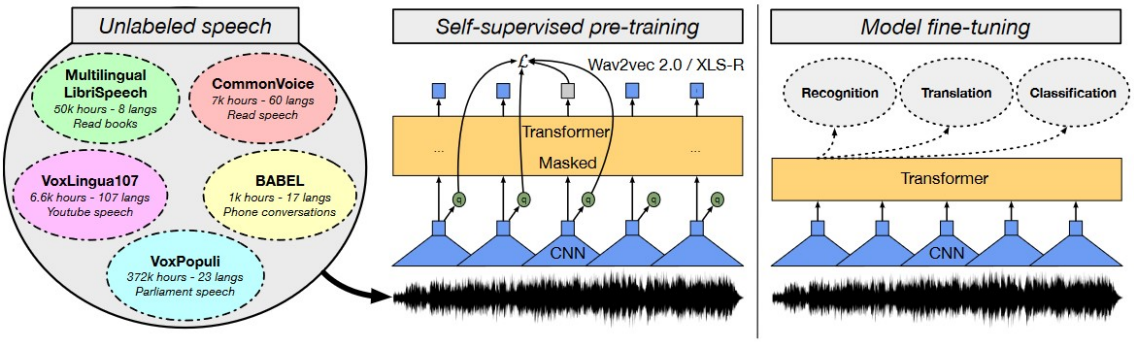


Figure 2.13: Architecture of Wav2Vec2-XLS-R, figure taken from [37]

The architectural design is equal to that of the Wav2Vec 2.0 model, with the notable difference being the corpora used for pre-training, as seen in Figure 2.13. 128 different languages in total are present in the dataset, compared to the previous 53 in the XLSR-53 model. These languages occur in varying degrees of size which have been categorised by the authors into high-resource (>1k hours), mid-resource (>100 hours) and low-resource (< 100 hours) and which can be seen in figure 2.14. To combat this apparent unequal distribution, they used a sampling distribution during training which first upsampled languages inside a corpus and then upsampled the corpus itself by treating them like languages. [37]

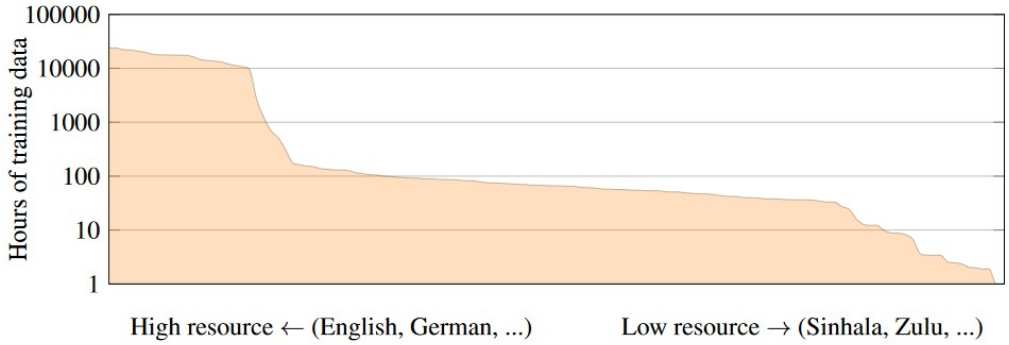


Figure 2.14: Distribution of languages used during training in the Wav2Vec2-XLS-R model, figure taken from [37]

As outlined in section 2.5 this model will be used as the basis for all of the experiments done in this thesis.

2.6 Language Models

Language models (LM) are an integral part of NLP by providing a mechanism that, based on statistical and probabilistic methods, can give a probability that a sequence of words is valid. In this context, validity does not necessarily mean grammatical validity, but that the word sequence is valid on how a specific text could be written, originating from the corpora that was used during training of an LM, which can cause varying degrees of accuracy. Using LM during or after training has demonstrated to be very beneficial for the predictions made by a model. Based on the appendix C in the original Wav2Vec2 paper [16] this is also true for the models used in this thesis.

Several implementations exist to create a LM like unigrams or a bi-directional approach. This thesis only uses the so-called n-gram approach, where n stands for the number of words, letters, syllables, or phonemes in a sequence for which a probability is calculated. Meaning that a 5-gram word model will look at word sequences of 5, for example: "I love green tea with" and then "love green tea with honey", and calculate the probability of the next word. Formalised, an n-gram model predicts the probability P of a word x_i based on the previous n words. [47]

$$P(x_i|x_{i-(n-1)}, ..., x_{i-1}) \tag{2.6}$$

In the context of the previous example "honey" would be predicted by using the sentence "I love green tea with" resulting in probability 2.7. Important to note is, that the initial predictions provided by the preceding CTC model will not be perfect as shown in this example, but will have missing letters or entire words resulting in something like "I lov green tea wif", which the LM then tries to fix by applying these probabilities. [47]

$$P(\text{honey}|\text{I}, \text{love}, ..., \text{with}) \tag{2.7}$$

Words in an n-gram are never taken out of order to contain the accurate meaning of the sentence for comparison. The reason for excluding the other methods is the KenLM library [48] that will be used for training of the LMs in this thesis, because of its fast and relatively cheap memory resource cost. [47]

2.7 Evaluation

Multiple evaluations metrics have been developed with each giving specific information about its application area. Several metrics have to be introduced, as this thesis will apply both Speech-To-Text translations (STT) and classification tasks.

2.7.1 Speech-To-Text (STT)

STT concerns itself with text-based evaluation for which two specific metrics are relevant, namely word error rate (WER) and bilingual evaluation understudy (BLEU). Important to note is that the resulting values of these metrics can vary greatly based

on the data and target languages used during training. High-resource languages like German and English consistently achieve good to very good scores, while low-resource languages or dialects still fall far behind.

Word Error Rate (WER)

First is the word error rate, which works on a word level using the Levenshtein distance and calculates a value between 0 and 1 by looking at the number of errors made by the model and divides them with the total number of words. 0 means a perfect alignment with the true label, while 1 means that there is no alignment. Equation 2.8 formalises this with the dividends as errors containing substitutions S, deletions D and insertions I with the total number of words N as divisors.

$$WER = \frac{S + D + I}{N} \tag{2.8}$$

BLEU

BLEU [49] has become the de facto standard for STT evaluation tasks and is thus an important metric to discuss. It is based on the comparison of n-grams which have been explained in section 2.6. BLEU is calculated by multiplying the Brevity Penalty with the Geometric Average Precision as seen in equation 2.9 and results in a value between 0 and 1 where, in contrast to WER, 0 is the worst possible score and 1 a perfect score.

$$BLEU = BrevityPenalty * GeometricAveragePrecision(N) \tag{2.9}$$

The Geometric Average Precision is calculated by looking at N different n-gram probabilities with an uniform weight $w_n = 1/N$. $N = 4$ is typically used, which means that the predictions made by the model are each looked at in an unigram, bigram, trigram and 4-gram setting. They are then compared with the truth label using a modified precision p_n that is the sum of the clipped n-gram counts divided by the total number of predicted words, formalised in equation 2.10.

$$\begin{aligned} GeometricAveragePrecision(N) &= exp(\sum_{n=1}^{N} w_n * log(p_n)) = \prod_{n=1}^{N} p_n^{w_n} \\ \text{where } p_n &= \frac{\sum_{C \in \text{Candidates}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \text{Candidates}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')} \end{aligned} \tag{2.10}$$

The last element is the Brevity Penalty which penalises sentences that are too short, to remove a possibility where unigrams could influence the predictions when a it is just "tea" or "honey", which could give a perfect score of 1 / 1 = 1. By using a parameter predicted length c and target length r, the penalty ensures that the value is never over 1. If the predicted sentence is too short compared to the target, the penalty will lower the value and thus lowering the BLEU score. A predicted length

of $c = 10$ and target length $r = 10$ will cause a Brevity Penalty of 1, as seen in equation 2.11.

$$BrevityPenalty = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c <= r \end{cases} \tag{2.11}$$

The interpretations of BLEU are often difficult to grasp, as objective good results appear between 0.6 and 0.7 while a score of 1.0 is often a sign of overfitting instead of a good score. Meaning the system could not be used in a different setting with never before seen data. BLEU is often normalized to the scale of 0-100 because it is not a percentage based metric. Table 2.1 summarises the interpretation of the scores.

BLEU (normalized)	Interpretation
< 10	Almost useless
$10 - 19$	Hard to get the gist
$20 - 29$	The gist is clear, but has significant grammatical errors
$30 - 40$	Understandable to good translations
$40 - 50$	High quality translations
$50 - 60$	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

Table 2.1: BLEU interpretation, table taken from [50]

2.7.2 Classification

Classification uses a set of evaluation metrics that aim at giving insight into the different classes that are being classified by the model. In this thesis, F1-Score will be the main evaluation metric as there are no binary class setups which would benefit from the accuracy metric.

Precision and Recall

Precision or confidence is used to provide insight into the proportion of positive identifications that were correctly classified by the model. The metric returns a value between 0 and 1, where 1.0 would be a perfect score as formalised in 2.12. [51]

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2.12}$$

Recall is used to describe the proportion of actual positives that were identified correctly which can be seen in equation 2.13.

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2.13}$$

F1-Score

Precision and recall are only relevant when taking both into account at the same time. However, as they are often in an inverse relationship to each other, a different metric has to be used - the F1-Score. It uses a harmonic mean over precision and recall to combine the two metrics into one by definition of equation 2.14. [51]

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.14}$$

Macro F1-Score

The above-mentioned equation is only valid in a binary setup. Multi-class setups need a so-called Macro F1-Score, which is calculated by using a macro average precision, equation 2.15, and a macro average recall, equation 2.16, where K is the amount of classes present during the calculation. The formula for precision and recall stay the same as in equations 2.12 and 2.13 but are summed up and then divided by the K classes. [51]

$$\text{MacroAveragePrecision} = \frac{\sum_{k=1}^{K} Precision_k}{K} \tag{2.15}$$

$$\text{MacroAverageRecall} = \frac{\sum_{k=1}^{K} Recall_k}{K} \tag{2.16}$$

The macro F1-Score is then calculated by using these averages in the formula 2.17.

$$\text{Macro } F1 = 2 * \frac{\text{MacroAveragePrecision} * \text{MacroAverageRecall}}{\text{MacroAveragePrecision} + \text{MacroAverageRecall}} \tag{2.17}$$

Weighted F1-Score

Since multi-class setups can have unequal data distribution, a third F1-score has to be introduced. This is the weighted F1, which takes the size of each class into account during the calculation of Precision and Recall. The number of samples is denoted as n_k in equation 2.18 and 2.19. [51]

$$\text{WeightedAveragePrecision} = \frac{\sum_{k=1}^{K} Precision_k * n_k}{\sum_{k=1}^{K} n_k} \tag{2.18}$$

$$\text{WeightedAverageRecall} = \frac{\sum_{k=1}^{K} Recall_k * n_k}{\sum_{k=1}^{K} n_k} \tag{2.19}$$

The F1 is then calculated using the weighted precision and weighted recall in the same manner as the default F1 or macro F1.

$$\text{Weighted Average } F1 = 2 * \frac{\text{WeightedAveragePrecision} * \text{WeightedAverageRecall}}{\text{WeightedAveragePrecision} + \text{WeightedAverageRecall}} \tag{2.20}$$

Chapter 3

Data

3.1 Overview

The data used for the pre-training and experiments stem from various sources. Corpora that are specifically created for ASR purposes are labelled and can be used during fine-tuning. While others, such as parliamentary data, are simply recordings of proceedings for documentation purposes and may be accompanied by written reports. However, if they are to be used as labelled data, they have to be further processed, for example by breaking the audio down into individual sentences and cutting the recordings in a manner so that they are aligned with the texts. Not publicly available data or data that would need special licensing were provided by the Centre for Artificial Intelligence (CAI) at ZHAW. The distribution of the different datasets is illustrated in Figure 3.1

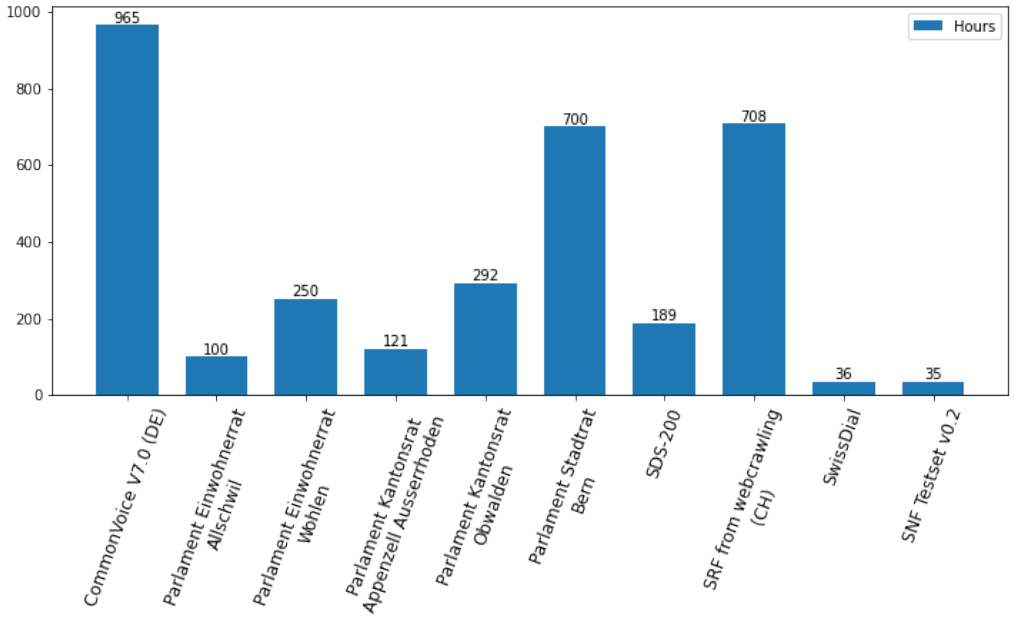


Figure 3.1: Speech hours per dataset

A minor side task of this work is to collect speech recordings from open sources such as TV and radio broadcasts or parliamentary reports. The aim is to increase the amount of unlabelled data available to pre-train ASR models. To this end, data from the Schweizer Radio und Fernsehen (SRF) from webcrawling (CH) collection were extracted with a web crawler from the website of the SRF [52] broadcaster.

3.2 Corpora

CommonVoice [53] is a collection of transcribed speeches containing 93 languages. With a total of more than 20'000 hours (validated nearly 15'000) of recordings. The German language in the V7.0 corpus consist of 965 validated hours and is the largest dataset used in this thesis. Metadata, besides the text of the sentence itself, include age group, gender and accent.

The dataset **Parlament Einwohnerrat Allschwil** comprises 100 hours of audio recorded in the local council of the municipality of Allschwil BL, with a predominance of the Basel dialect.

The **Parlament Einwohnerrat Wohlen** consists of approximately 250 hours of recordings of the local council of the municipality of Wohlen AG. The most widely spoken dialect in the dataset is that of AG and accompanied by the recordings there are also official transcripts/reports.

Parlament Kantonsrat Appenzell Ausserrhoden is a collection of recordings of the cantonal council of Appenzell Ausserrhoden with a linguistic predominance of the AR dialect. It includes approximately 120 hours of speeches that are automatically aligned to Standard German text data on sentence-level.

The dataset **Parlament Kantosrat Obwalden** includes just over 290 hours of audio recorded in the Obwalden cantonal council mostly in its OW dialect. The speeches are automatically aligned Swiss German to Standard German text data on sentence-level.

The **Parlament Stadtrat Bern** is a dataset with recordings from the Bern City Council. It consists of 700 hours of speeches with a predominance of the BE dialect. The speeches are automatically aligned Swiss German to Standard German text data on sentence-level.

The **SDS-200** (Schweizer Dialektsammlung) [5] is a dataset containing several Swiss German dialects with their respective transcriptions in standard German. It is the first Swiss German corpus that contains speech samples from all over Switzerland with a data distribution that generally represents the Swiss population (see figure 3.2). Outliers are the cantons of ZH and VS which are over-represented.

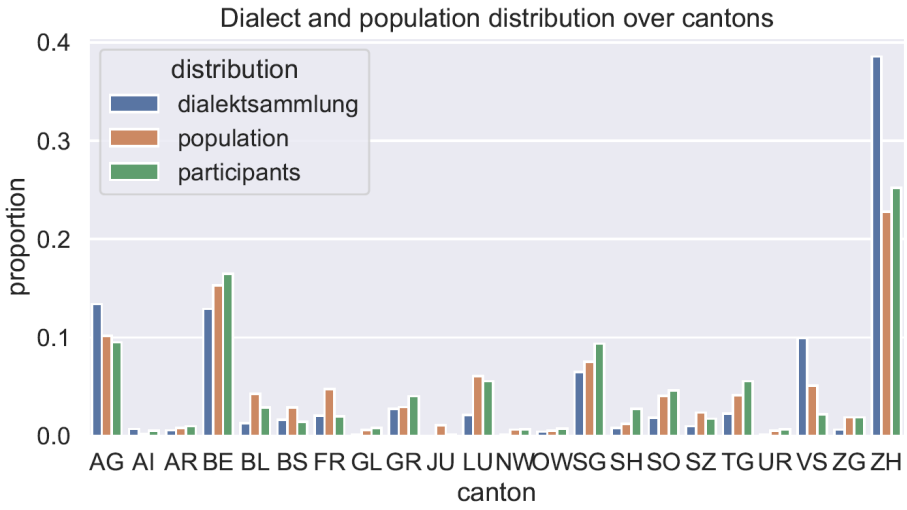


Figure 3.2: Figure taken from [5]

It consists of raw data (188.9 hours) and filtered data (178.3 hours). The filtered data is passed through a validation process to ensure high quality, while the raw data remains in the dataset in case different filtering criteria are to be applied. The dataset already provides splits for validation (5.2 hours) and testing (5.4 hours). These are selected to ensure that only validated data are present in the validation/test splits and to have a good variety, speakers with between 5 and 200 recordings. It is also guaranteed that speakers are only present either in the training splits or in the validation/test splits. In addition to the transcription in standard German and the dialect, the age group and gender of the speaking person are also indicated in the metadata.

SRF from webcrawling - Around 3000 hours of podcasts were extracted from the SRF website [52] using webcrawling that automatically downloads publicly available data from the website. The audio was then assigned into three different categories according to the language spoken in the podcast: CH, DE and Mixed (CH+DE). The separation was done manually by skipping through up to 10 random episodes of each podcast to ensure that they could generally be assigned to one of the three categories. DE and Mixed have 401 and 1812 hours of audio respectively, while the most important CH category comprises 741 hours. Of these three, only the data in CH were used for the experiments in this thesis. Before the samples could be used in the pre-training setup, they had to be pre-processed to remove unwanted parts of the podcasts such as music, noise or special effects; and thus only obtain audio containing people speaking. To do this a script was written which, given a folder of raw files, starts by converting the audio format from MP3 to WAV. Each file is then sent to Spinningbytes' internal Voice Activity Detection service which returns a .json file indicating when someone is actually speaking. The script then cuts the parts with the voice out of the .wav file and breaks it down further into segments between 2 and 10 seconds. This operation cut approximately 32 hours in the CH dataset resulting in 708 hours of usable audio.

The **SwissDial** [54] was created by ETHZ and consists of 36 hours of audio recorded in 8 different Swiss German dialects (AG, BE, BS, GR, LU, SG, VS, ZH). All dialect share the same German sentences, which had to be translated into their respective dialect. The metadata thus includes the same sentence several times, in both Standard German and translated into the dialects for which it was actually recorded. Additionally, the topic of the sentence is provided.

The **SNF Testset v0.2** consists of nearly 35 hours of recordings in the different Swiss-German dialects and their respective transcriptions. In addition to the canton, the metadata also includes, but is not limited to, sub-region, age and gender.

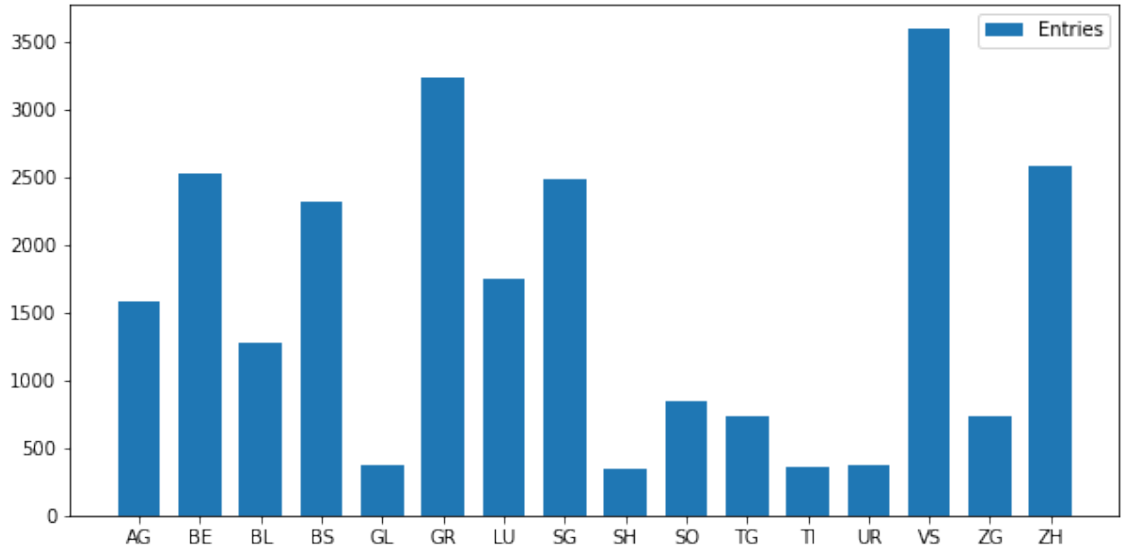


Figure 3.3: Entries per cantons of the SNF Testset v0.2 dataset

Chapter 4

SwissText Shared-Task

While writing this thesis, the opportunity presented itself for us to partake in the SwissText hosted "2nd Swiss German Speech to Standard German Text" shared task [6]. We were invited by our supervisors to enter the best performing models into the competition to test their capabilities in a competitive setting.

4.1 Objective

The aim of the shared task is to create a system that can translate Swiss German dialects into Standard German. Evaluation is conducted on a special dataset containing 5 hours of the Graubünden dialect. Restrictions are in place concerning the corpora allowed to be used during training for a fair contest, which are listed below:

- SDS-200 [5]
- SwissDial [54]
- CommonVoice (DE, FR, IT) [53]
- Evaluation dataset

All corpora have been described and analysed in Chapter 3. For CommonVoice the three languages of German, including a small subset of dialects, French, and Italian are permitted to be used.

4.2 Evaluation

BLEU is used as the evaluation metric (see section 2.7) and the data is separated between a public and private split, with the private split only being revealed after the submission deadline. The team achieving the highest score on the private split wins the competition. Both splits contain 50% of the evaluation data to remove the possibility of participants overfitting on the public split. Evaluation vocabulary is limited to numbers and lowercase characters. The organizers supplied a baseline model that has to be beaten. The BLEU score of this model on the public split is 0.7044.

Chapter 5

Experimental Setup

5.1 Objective

As mentioned in Chapter 1, the thesis and its experiments aim to determine the impact of adding additional pre-training data to an existing pre-trained model for a specific language or dialect group. We specifically want to test this thesis by applying a multitude of Swiss German datasets to the Wav2Vec2-XLS-R model. By submitting our results to the second Swiss German translation shared task hosted by SwissText [6], we seek to test the model's capabilities in a competitive setting.

Allowing for a more complete overview of the model's abilities, we apply two different categories of experiments: STT translation and speech classification. For STT translation, we want to determine if the translation of the different Swiss German dialects into Standard German improves. Classification aim at understanding the different relations of the dialects and compare the results the our previous work done in the scope of a project thesis. The impact of the pre-training will be explored as well.

5.2 Infrastructure

The primary infrastructure provided by ZHAW includes four GPU instances on the ZHAW APU OpenStack cluster, with each instance having access to 16GB RAM, 8 vCPUs, and an NVIDIA Tesla T4 with 16GB DDR6. A total of 5TB was allocated for data. For technical reasons, it was not possible to use multiple graphics cards together for experiments. Some tasks were carried out on the DGX-1 (NVIDIA Tesla V100-SXM2-32GB) and DGX-A100 (NVIDIA Tesla A100-SXM4-40GB) servers, to which we did not have direct access and therefore had to pass through our secondary supervisor. The addition of A-100 instances was a significant milestone during this thesis as the training duration of our pre-train models was reduced by up to 90% from initially 8 months to just 3 weeks.

5.3 Corpora

We aim at utilizing all in Chapter 3 mentioned Swiss German corpora in their respective area of application, as shown in Table 5.1. One exception is the CommonVoice corpus for German, which is used for an experiment where both Swiss German and Standard German speech are used to pre-train a model. Looking at the column "Purpose", one can identify if a corpus is unlabelled or labelled. STT translation and classification apply labelled corpora while pre-training uses unlabelled corpora.

Dataset Name	**Language**	**Dialect**	**Hours**	**Purpose**
CommonVoice v7.0 (DE)	DE	None	965	Pre-Train
Parlament Einwohnerrat Allschwil	CH	Mainly BL/BS	100	Pre-Train
Parlament Einwohnerrat Wohlen	CH	Mainly AG	250	Pre-Train
Parlament Kantonsrat AR	CH	Mainly AR	121	Pre-Train
Parlament Kantonsrat OW	CH	Mainly OW	292	Pre-Train
Parlament Stadtrat Bern	CH	Mainly BE	700	Pre-Train
SRF (CH)	CH	Mixed	708	Pre-Train
SDS-200	CH	Mixed	189	STT & Classification
SwissDial	CH	Mixed	36	STT
SNF Testset v0.2	CH	Mixed	35	STT
Total hours used			**3482**	

Table 5.1: Corpora used for training

A special SDS-200 split was created for the classification task using a setup that we used for the project thesis, which closely imitates the original split performed by the authors. It was necessary, because the original split was done with STT in mind instead of classification and as such had samples with no canton in their metadata. We applied a 80/20 split for training and test data with samples that had a canton assigned to them. Speakers were only presented in either test or train and a general validity control was applied with users having a mean quality lower than 0.5 being dismissed. Samples with no quality were still included to allow a certain degree of real-world applicability where audio can have varying degrees of quality.

Important to note is that the SwissDial corpus [54] was only added at the beginning of May 2022 when we were invited to participate in the SwissText shared task. All trainings started before that time frame thus do not contain any data from this corpus.

5.4 Pre-Processing

The in 5.3 discussed corpora have to be pre-processed for further use downstream in training. We employed the same strategy for all datasets to have a standardized pipeline during training when adding new data. This generally includes a conversion to the WAV audio format and re-sampling the audio to 16kHz. The data is then stored in the HDF5 hierarchical data format using the h5py Python library [55],

with which vast reductions in memory usage can be achieved. Some differences exist between labelled and unlabelled corpora which will be addressed now.

5.4.1 Labelled

The transcripts of the labelled data had to be stored in a way that is both efficient to read during training and structured for assignment to the correct sample. Sentences and classifications were thus written to metadata files which were read and parsed for their assigned samples during the respective training.

5.4.2 Unlabelled

The data used for pre-training was often not in a format usable for training. Most audio files were provided as podcasts or meetings which last between 30 minutes and multiple hours. Reading multiple such files would lead to a memory overflow during training. As such, the data had to be cut into audio samples of between 2 and 10 seconds. The cuts were performed using Voice-Activity-Detection and did not adhere to sentence structures, so the resulting audio samples were sliced at random points during a speech.

5.5 Model Selection

Based on the promising results of the Wav2Vec2-XLS-Rpaper [37], a decision was made to use this model. Multiple XLS-R versions exist, namely the 300M, 1B, and 2B models. The numbers refer to the amount of trainable parameters each model possesses. Training duration scales based on these parameters, with the smallest 300M having the fastest and the largest 2B model the slowest training. Concerning the limited time frame in which this thesis has to be completed, the decision was made to primarily use the 300M model. During the writing of the thesis, an opportunity arose to apply the 1B model in a limited framework as well, owing to the section 5.2 described addition of A-100 GPUs for further computational power. The models are available through the transformer python library [56] hosted by HuggingFace.

The models were released in November 2021 and compared to its XLSR-53 predecessor, only a fraction of research has been released thus far. As such, not many fine-tuned models are available on HuggingFace [56] for further downstream tasks, like the ones applied in this thesis. Owing to this, only the base models can be considered being used.

For the classification task, a reference model that is based on the XLSR-53 architecture will be utilized. As outlined in section 5.1, this is done because this dissertation is a continuation of a project thesis. By comparing the model to our results, we hope to gain valuable insight into the capabilities of the XLS-R model and our additions during the pre-training task.

5.6 Language Model

Two models based on the KenLM library [48] are applied, with both having a different purpose. The first model "LM-Wiki" is a 5-gram model where n-grams are pruned if they occur less than 5 times and is trained on a German Wikipedia corpus. The second "LM-CC-100" model is based on a subset of the German CC-100 corpus [57][58] and is a 5-gram model that prunes n-grams occurring less than 3 times. The reason for the difference in pruning is the increased size of the LM-CC-100 model, which would have increased even further if pruning of less than 5 was to be applied. An additional difference is the vocabulary used in these models. While the LM-Wiki uses punctuations, numbers, and characters that are both upper- and lowercase, the LM-CC-100 only comprises numbers and lowercase characters. This separation was needed for the SwissText shared task as described in Chapter 4.

Several hyperparameters have to be defined during evaluation with the LM. Based on tests with multiple models, a decision was made to use the values listed in Table 5.2. Reranking is done using a German GPT2 model. Some parameters have to be explained:

- **Hyp rerank**: Amount of top-n hypotheses the GPT-2 model reranks
- **Beam size**: Size of beam search width
- **Alpha**: weight for the LM during shallow fusion
- **Beta**: a constant weight for length score adjustment during scoring

Parameter	Value
Hyp rerank	200
Beam size	800
Alpha	0.5
Beta	1.0

Table 5.2: Language model evaluation parameters

5.7 Metrics and Evaluation

To evaluate the translation experiments, the in section 2.7 mentioned WER and BLEU metric are used. Classification experiments are evaluated according to the F1 score. These metrics allow for a sufficiently exhaustive method of comparing results. The tool used to keep track of the experiments is Weights & Biases [59], which has a convenient interface for displaying graphs, logs and metrics.

5.8 Training Details

This section aims to provide insight into the most important hyperparameters for each experiment and the reasoning why the specific values were chosen. Important to note is that parameters differ for the experiments based on both infrastructure, models, and standard values defined by the NLP community.

5.8.1 Pre-Training

For pre-training, the most important difference is the batch size and gradient accumulation with which the amount of data a GPU is processing at any given time can be controlled. We aimed at applying 1h of audio data per GPU, which corresponds to the values seen in table A.2.

Instead of using training epochs as a fixed training duration, we defined a value to be large enough that our goal of 120k to 150k global steps could be reached. The reason for this specific amount of steps is a blog post [60], on which our code base relies on, by the HuggingFace research engineer Patrick von Platen, where the same amount of global steps were applied. By imitating this number, we can draw a comparison between his and our training progress. All mentioned parameters are in Table A.2 in Appendix A.

5.8.2 STT Translation

As STT experiments were performed on weaker GPUs than the pre-training models, both the batch size and gradient accumulation steps had to be reduced to 4 and 8, respectively. The learning rate was set at $3e^{-5}$ with 100 warmup steps and 25 training epochs. The parameters can be viewed in Table A.3 in Appendix A

5.8.3 Classification

Compared to the STT experiments, the save steps and evaluation steps parameter changed from 2000 to 1000 and 1000 to 500, respectively. The reason for this is the reduced training duration of the classification tasks because labelled data is available. The values are displayed in Table A.4 in Appendix A.

5.9 Trainings

This section will summarize the different trainings performed in this thesis. By explaining the setups, the planned comparisons, and the corpora used for each of them, a sufficient understanding should be able to be achieved.

5.9.1 Pre-Training

As outlined in Chapter 1 and section 5.1, we aim to apply and measure the impact of additional pre-training data on the Wav2Vec2-XLS-R architecture. Most down-

stream tasks in STT and classification rely on these additions before their respective training can be started.

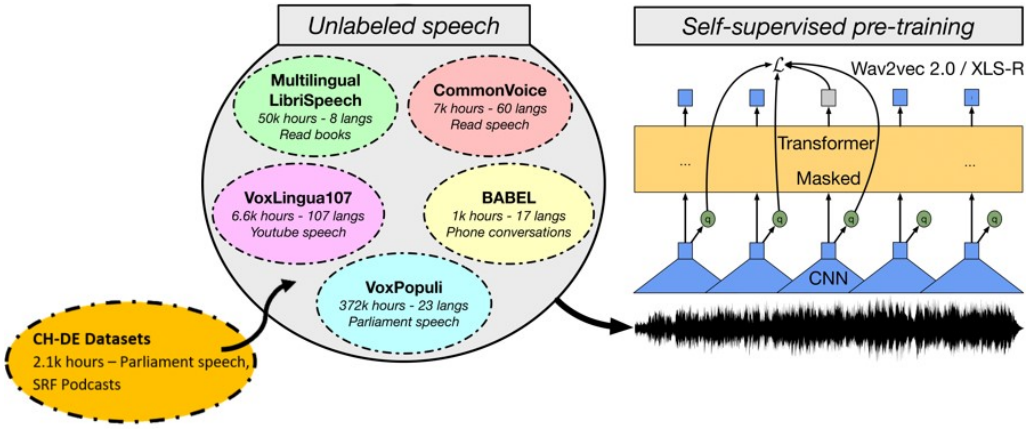


Figure 5.1: Addition of Swiss German data to the Wav2Vec2-XLS-R architecture

Three models are trained for this task, comprising two 300M models and one 1B model. The difference in the XLS-R-300M's is the addition of Standard German data to the pre-training for one of them. This should provide an additional option for comparison in the downstream tasks. Both the 300M "CH-Pretrain-300M" and 1B "CH-Pretrain-1B" Swiss German models will apply 2171 hours of Swiss German data, as visualized in 5.1, while the training for the mixed CH-DE 300M model "CH_DE-Pretrain-300M" will comprise 3206 hours. Both 300M models were first trained on the DGX-1 cluster before being moved to the A-100 instances for faster training. Figure 5.2 provides an extensive overview of the three models.

	Category			Infrastructure			Model		Based on		Train Dataset										Eval Dataset	
	Pretrain	STT	Classification	APU	DGX-1	DGX-A100	300M	1B	Base	Pretrain	CommonVoice (DE)	Parlament Einwohnerrat Allschwil	Parlament Einwohnerrat Wohlen	Parlament Kantonsrat AR	Parlament Kantonsrat OW	Parlament Stadtrat BE	SRF from webcrawling (CH)	SDS-200 (train)	SwissDial	SDS-200 (test)	SNF Testset v0.2
CH-Pretrain-300M	x				x	x	x		x			x	x	x	x	x	x				
CH-Pretrain-1B	x					x		x	x			x	x	x	x	x	x				
CH_DE-Pretrain-300M	x				x	x	x		x		x	x	x	x	x	x	x				

Figure 5.2: Setup for pre-train models

The training will use the "Wav2Vec2ForPreTraining" class of the transformer library [56] and is based on the code base of Patrick von Platen at HuggingFace [60].

5.9.2 STT Translation

The primary evaluation mechanism for the pre-trained models are the STT translation experiments from Swiss German into Standard German. The reason for this translation is that Swiss German does not adhere to any grammatical or vocabulary standards. Speakers of this language freely choose on how they write and speak based on the region they grew up in. After translating Swiss German speech to Standard German we can then apply the set of grammatical rules set out by Standard German to standardize the dialects into one form. This enables us to both compare the dialects better and to use existing German systems that have been optimized throughout the years for a better prediction. The "Wav2Vec2ForCTC" class of the transformer library [56] will be used for this task. By adding a KenLM model to the fine-tuning process, we aim to improve the translations.

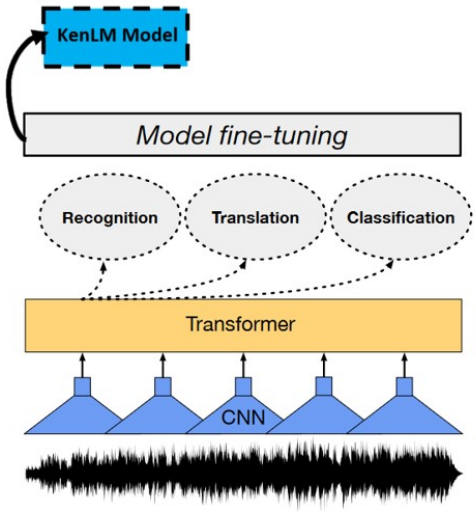


Figure 5.3: Addition of KenLM models to fine-tuning to improve translations

Multiple translation experiments will be performed, as illustrated in Figure 5.4. An explanation has to be given to understand the naming convention of the different experiments. As an example "CH-STT-FromPretrain-300M-75k" is based on the Swiss German 300M pre-train model and uses a checkpoint at 75k global steps which corresponds to 50% of the training progress.

Two additional models were added for the SwissText task, denoted with "Limited-Vocab", which only used lowercase characters and numbers for their prediction to match the allowed characters set out by the taskmasters [6]. Owing to this, the LM-CC-100 will be used instead of the LM-Wiki, as outlined in section 5.6

	Category			Infrastructure			Model		Based on		Train Dataset									Eval Dataset	
	Pretrain	STT	Classification	APU	DGX-1	DGX-A100	300M	1B	Base	Pretrain	CommonVoice (DE)	Parlament Einwohnerrat Allschwil	Parlament Einwohnerrat Wohlen	Parlament Kantonsrat AR	Parlament Kantonsrat OW	Parlament Stadtrat BE	SRF from webcrawling (CH)	SDS-200 (train)	SwissDial	SDS-200 (test)	SNF Testset v0.2
CH-STT-Base-300M		x		x			x		x									x			x
CH-STT-Base-300M-LimitedVocab		x			x		x		x									x	x	x	x
CH-STT-FromPretrain-300M-75k		x		x			x			x								x			x
CH-STT-FromPretrain-300M-Full-1		x		x			x			x								x			x
CH-STT-FromPretrain-300M-Full-2		x		x			x			x								x	x		x
CH_DE-STT-300M-FromPretrain-Full		x		x			x			x								x	x		x
CH-STT-Base-1B		x				x		x	x									x	x	x	x
CH-STT-Base-1B-LimitedVocab		x				x		x	x									x	x	x	x
CH-STT-FromPretrain-1B-75k		x			x			x		x								x	x	x	x
CH-STT-FromPretrain-1B-Full-1		x				x		x		x								x	x	x	x

Figure 5.4: Setup for STT models

Because of the limited time frame and computational power, various experiments had to either be removed or reduced in their capacity. Owing to this, only the "300M-Full" model could be repeated to reduce evaluation variation error. We recognise, however, that these are not sufficient repetitions to fully remove said error. Additionally, the CH_DE model could only be used once in STT. In the same essence, multiple models only evaluated on the SNF corpus to reduce training duration.

5.9.3 Classification

The classification task is based on an experiment performed in the project thesis "Automatic Detection of Swiss German Dialects using Wav2Vec" of which this thesis is a continuation. This setup achieved the highest evaluation score with a macro F1-score of 45.95% by grouping the cantons based on their regional proximity and linguistic similarity.

Analysis of the different dialects in [61] showed that the "Ostschweizer Dialekte" in the east of Switzerland were historically more closely related to each other than to the dialects in the central region. The central cantons of Aargau, Lucerne, Zug and Zurich have been described as "Übergangsmundart Kantone" in which influences of both eastern and western dialects can be found. A separation was discovered based on the so-called "Brünig-Napf-Reuss" line shown in Figure 5.5, which separates east from west. Western dialects work similarly to the eastern dialects in that they are closely related to each other and can be grouped for classification.

Figure 5.5: Brünig-Napf-Reuss line in red, High-Alemannic area in yellow, figure taken form [62]

The last group contains cantons of both the "Innerschweiz" to which Glarus, Schwyz, Uri, et al. belong and the special Highest-Alemannic dialect found in the canton of Valais. All these cantons have low amounts of data available for training and were thus grouped together. While this may not be the most beneficial, owing to the Valais dialect, which is not related to the dialects found in the Innerschweiz, they still had to be clustered into a class that had enough data to support them during the classification task.

Four distinct groups are thus created, shown in Figure 5.6, with the Eastern-High-Alemannic dialects (EA) in blue, the Central-High-Alemannic dialects (CA) in yellow, the Western-High-Alemannic dialects (WA) in green and the Highest-Alemannic dialects (HA) in red.

Figure 5.6: Four regional dialect groups

Table 5.3 provides insight into the data distribution of the different clusters. As mentioned above, the red HA group has the lowest amount of resources available for training. The CA region comprises around 50% of all available data owing to the fact that Zurich is the most populous canton in Switzerland and, as outlined in Chapter 3, attributes to 40% of all samples in the SDS-200 corpora.

Nr.	Name	Abbrev.	Cantons	Training sample size
1	Highest-Alemannic	HA	GL, NW, OW, SZ, UR, VS	5814
2	Western-High-Alemannic	WA	BE, BS, BL, FR, SO	19237
3	Central-High-Alemannic	CA	AG, LU, ZG, ZH	37967
4	Eastern-High-Alemannic	EA	AI, AR, GR, SG, SH, TG	12465

Table 5.3: Regional dialect groups definition

The in Figure 5.7 shown experiments are similar to the STT experiments with the exception of a model that uses one of the STT models as its base, denoted as "FromSTTPretrain". It allows us to make a more valid comparison with the XLSR-53 model from the project thesis. There, a on Swiss-German fine-tuned STT model, provided by the University of Applied Sciences Northwestern Switzerland (FHNW) [22], was used as the base model, which at the time improved the F1-score by up to 15%.

	Category			Infrastructure			Model		Based on		Train Dataset									Eval Dataset	
	Pretrain	STT	Classification	APU	DGX-1	DGX-A100	300M	1B	Base	Pretrain	CommonVoice (DE)	Parlament Einwohnerrat Allschwil	Parlament Einwohnerrat Wohlen	Parlament Kantonsrat AR	Parlament Kantonsrat OW	Parlament Stadtrat BE	SRF from webcrawling (CH)	SDS-200 (train)	SwissDial	SDS-200 (test)	SNF Testset v0.2
CH-Classify-Base-300M-1			x	x			x		x									x		x	
CH-Classify-FromPretrain-300M-Full-1			x	x			x			x								x		x	
CH-Classify-FromPretrain-300M-Full-2			x	x			x			x								x		x	
CH-Classify-FromSTTPretrain-300M-Full			x	x			x			x								x		x	

Figure 5.7: Setup for classification models

The code implementation uses the "Wav2Vec2ForSequenceClassification" class in the transformer library [56] and employs a dimensionality of 1024 for the encoder and projection layer.

Chapter 6

Results

This chapter analyzes the results gained by the pre-training, the STT translation and classification experiments, and the SwissText shared task. The results of the STT translation were the primary factor on which the impact of pre-training was examined. The classification was used for comparison between XLSR-53 and XLS-R with the intention to gain insight into dialect identification. We will discuss the main findings and assumptions further in Chapter 7.

6.1 Pre-Training

No evaluation metric could be used for pre-training, as the data used for training was unlabeled. The loss function was taken instead as a substitution to analyze if the model learned something during training. By comparing this graph to the loss function of an example training by HuggingFace [60], we assured ourselves that the training moved in the correct direction. Training duration for the models lay between 22 and 29 days, with the 1B model taking the longest, relative to the global step. The 300M models could have finished faster, but as they were first trained on DGX-1 before being moved to the much faster A-100 instances, the duration was pushed further up.

Model	Training duration	Global steps
CH-Pretrain-300M	29d 8h	130k
CH-Pretrain-1B	22d 14h	112k
CH_DE-Pretrain-300M	24d 18h	154k

Table 6.1: Training duration of the different pre-train models

By comparing Figure 6.1 and Figure 6.2, we can discern that the models moved in the same slight downward direction, which meant our implementation was successful and the model was learning. The in section 2.5.7 discussed contrast and diversity loss functions are illustrated as well. Important to note is that the example of HuggingFace did not utilise an XLS-R model, but the monolingual Wav2Vec2-Large model. The Figure can thus not be compared directly to our loss function but has to be seen as a reference.

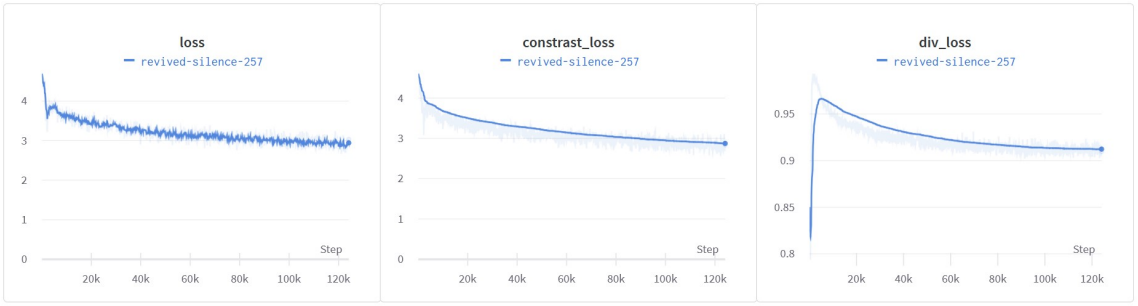


Figure 6.1: Loss functions of HuggingFace example, figure created by Patrick v. Platen using W&B, taken form [60]

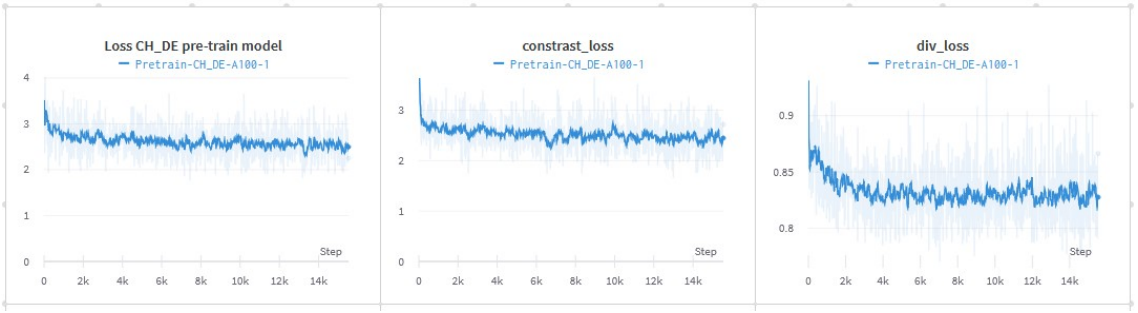


Figure 6.2: Loss functions of the CH_DE-Pretrain-300M model with smoothing factor 0.8, figure created using W&B

6.2 Experiments

6.2.1 STT Translation

The translation experiments were the primary evaluation factor for the impact pre-training has on the XLS-R model. We will separate the analysis for the 300M and 1B models to display their capabilities. In each section, a figure will be presented where the bold score represents the best BLEU for the given evaluation corpus compared to the other models.

300M Models

300M models were primarily trained on the APU instances and each took around 2 weeks to train. The only exception was the LimitedVocab model, which was trained on the DGX-1 and additionally evaluated on the SNF corpus. The results are illustrated in Figure 6.3 and show that the base model generally beat all other models. In the training evaluation, it reached the best score on the SNF with a WER of 25.08% and a BLEU of 0.5704 and in the SDS-200 evaluation with an LM a 21.56% WER and 0.6435 BLEU. The exception is the result of the LM SNF evaluation in which the Full-2 model achieved a WER of 20.17% and 0.6519 BLEU. This is surprising, considering that the Base model had a better evaluation on the

SNF during training. We assume the SwissDial corpus, which only the Full-2 and CH_DE model could take advantage of during training, had a positive influence on the model.

	Score SDS-200		Score SNF		LM		Score LM SDS-200		Score LM SNF	
	WER	BLEU	WER	BLEU	LM-Wiki	LM-CC-100	WER	BLEU	WER	BLEU
CH-STT-Base-300M	-	-	0.2508	**0.5704**	x		0.2156	**0.6435**	0.2047	0.6467
CH-STT-Base-300M-LimitedVocab	0.2654	**0.5624**	0.2710	0.5447		x	0.2247	0.6260	0.2217	0.6217
CH-STT-FromPretrain-300M-75k	-	-	0.2647	0.5470	x		0.2195	0.6344	0.2126	0.6341
CH-STT-FromPretrain-300M-Full-1	-	-	0.2682	0.5529	x		0.2211	0.6299	0.2160	0.6311
CH-STT-FromPretrain-300M-Full-2	-	-	0.2550	0.5624	x		0.2179	0.6394	0.2017	**0.6519**
CH_DE-STT-300M-FromPretrain-Full	-	-	0.2509	0.5674	x		0.2173	0.6361	0.2018	0.6489

Figure 6.3: Results of 300M models in speech translation, left side is the training evaluation, on the right the LM evaluation

This table also presents an additional finding with the CH_DE model achieving the second highest score on the SNF in both training and downstream LM evaluation. The experiment was only performed once and thus can not be used for a complete analysis of the impact Standard German has. Further experiments need to be performed to give a definitive statement, but we can assume that the impact should be positive based on the close linguistic relation that Swiss German and Standard German share.

The last finding is the apparent reduction in scores of the pre-train models compared to the base model, especially during training. When applying the Language Model the scores increase in different manners based on the training predictions. This would mean that pre-training has had a negative influence on the model instead of a positive one. Before analyzing this further, the results of the 1B have to be discussed.

1B Models

The 1B models were exclusively trained on the DGX-1 and A-100 instances to reduce training duration. They trained for around a week before finishing, which is a 50% reduction compared to the 300M models. The results depicted in Figure 6.4 show a similar outcome to the 300M models where pre-trained models were performing worse than their Base counterparts. Specifically, with 1B, the LimitedVocab model performed well over all stages of evaluation. It reached the highest score of all models in training, including the 300M experiments, with a 0.6249 BLEU on the SDS-200. The addition of LMs increased the scores on the SNF evaluation set to 0.6889 BLEU and a WER of 18.08%.

Surprising compared to the 300M models is that the LimitedVocab model there did not perform nearly as well as its 1B counterpart. This may be due to variation

and should be analyzed further. We assume for the time being that the reduction in vocabulary size helped the 1B-LimitedVocab during its predictions and thus had higher scores than the other experiments.

	Score SDS-200		Score SNF		LM		Score LM SDS-200		Score LM SNF	
	WER	BLEU	WER	BLEU	LM-Wiki	LM-CC-100	WER	BLEU	WER	BLEU
CH-STT-Base-1B	0.2282	0.6205	0.2222	0.6170	x		0.2008	0.6672	0.1875	0.6780
CH-STT-Base-1B-LimitedVocab	0.2257	**0.6249**	0.2182	**0.6236**		x	0.1928	**0.6805**	0.1808	**0.6886**
CH-STT-FromPretrain-1B-75k	0.2395	0.6008	0.2343	0.5976	x		0.2080	0.6533	0.1954	0.6609
CH-STT-FromPretrain-1B-Full-1	0.2485	0.5887	0.2435	0.5824	x		0.1956	0.6731	0.1859	0.6810

Figure 6.4: Results of 1B models in speech translation, left side is the training evaluation, on the right the LM evaluation

A different finding is the 1B experiments did not perform significantly better than their smaller 300M equivalent. We assumed that, based on the three times larger amount of trainable parameters, the model would reach much higher scores, but the difference is only around 4 BLEU. We currently assume that these are hyperparameter fine-tuning errors based on learnings during the SwissText conference, which will be discussed in section 6.3.

6.2.2 Classification

Comparison model

Before discussing the results of the XLS-R model, we have to provide context for the reference XLSR-53 model of the project thesis. The model used a STT fine-tuned model hosted on HuggingFace and provided by FHNW [22], which was then adapted to the classification task. During the evaluation, the model reached an accuracy of 52.89%, a macro F1-score of 45.95%, and a weighted F1-score of 0.5. The best group was the EA region with an F1 of 0.65, while the worst was the HA region with a score of 0.13. The predictions made by the model have been illustrated in Figure 6.5 and an overview of the scores are provided in Table 6.2

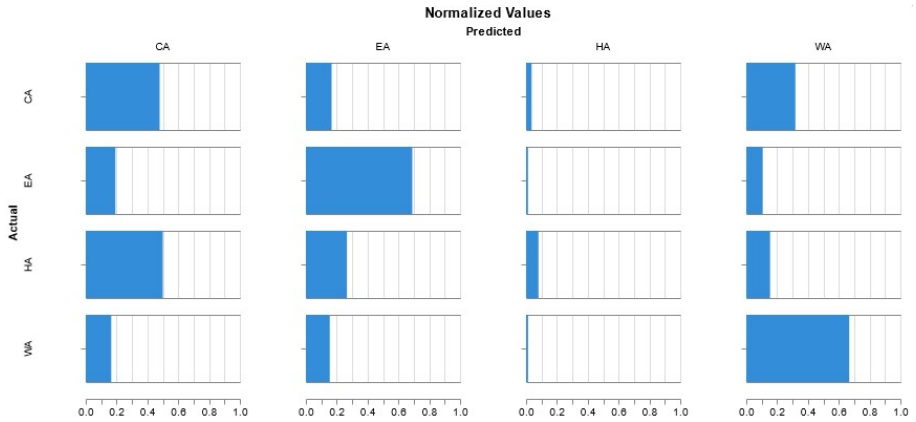


Figure 6.5: Confusion matrix of regional dialects in the project thesis

Region	Precision	Recall	F1-score
EA	0.62	0.69	0.65
CA	0.44	0.48	0.46
WA	0.53	0.67	0.59
HA	0.42	0.08	0.13

Table 6.2: Precision, Recall and F1-score of comparison model

XLS-R-based Models

Four experiments were run for the classification task with one base model as a comparison and three FromPretrain models. One of the pre-train models used an already fine-tuned model from the STT experiment as its base hoping to increase the evaluation scores, similar to the XLSR-53 comparison model. CH-STT-FromPretrain-300M-Full-1 had been selected as the base because it was the first of the two Full models to finish fine-tuning. However, as it also used SDS-200 for its training we suspected that the downstream classification training may overfit during training and had to be monitored as such. The models were exclusively evaluated on the SDS-200 test set containing 13644 samples. While the size of SDS-200 has increased since the project thesis, a general comparison should still be able to be made. The results are depicted in Figure 6.6.

	Score SDS-200		
	Accuarcy	F1 (Macro)	F1 (Weighted)
CH-Classify-Base-300M-1	41.75%	35.96%	38.00%
CH-Classify-FromPretrain-300M-Full-1	47.32%	40.51%	43.00%
CH-Classify-FromPretrain-300M-Full-2	48.52%	41.55%	45.00%
CH-Classify-FromSTTPretrain-300M-Full	52.45%	44.38%	49.00%

Figure 6.6: Results of the classification

The best model was CH-Classify-FromSTTPretrain-300M-Full which used the CH-STT-FromPretrain-300M-Full-1 fine-tuned Swiss German model as its base and achieved an accuracy of 52.45%, a macro F1-score of 44.38%, and a weighted F1-score of 0.49. The XLSR-53 comparison model could thus not be beaten by 1%. Contrary to the STT models, the base model was outperformed by all pre-train models by up to 9% macro F1 and 11% weighted F1. Based on these findings a closer inspection of the FromSTTPretrain model has to be made.

Looking at the results in Table 6.3 one can discern that some regions improved while others worsened. The already strong EA region increased to a F1 of 0.68 in the XLS-R experiment from the initial 0.65 in the XLSR-53 while the HA region decreased to 0.09 F1 compared to 0.13 by the reference model.

Region	Precision	Recall	F1-score
EA	0.70	0.66	0.68
CA	0.39	0.78	0.52
WA	0.63	0.40	0.49
HA	0.45	0.05	0.09

Table 6.3: Precision, Recall and F1-score of CH-Classify-FromSTTPretrain-300M-Full

Analyzing Figure 6.7 enables a better understanding of the false predictions the model made. The Eastern-Alemannic region has again largely been successful in being recognised by the model. Most erroneous predictions were due to the CA region, which makes sense considering the proximity of the two regions.

CA was most often confused with the WA region. This was prone to occur, considering the in section 5.9.3 outlined difficulty of separating the cantons of the central region into east and west. Achievements could be made here by splitting the dialects not based on cantons but on a zip-code level, which would allow for a better allocation to the east or west.

The scores of the WA worsened compared to the XLSR-53 model from 0.59 to 0.52, with an equal distribution of mistakes in both EA and CA. This is surprising because the regions of WA and EA are not near each other and are linguistically distinct.

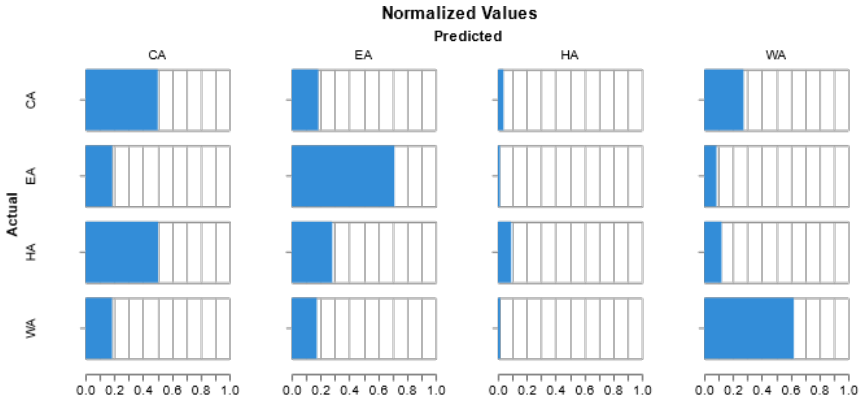


Figure 6.7: Confusion matrix of FromSTTPretrain model regional classification experiment

In last place is the HA region, which had the most difficult task of all regions considering the low sample size and the linguistic dissimilarity of the dialects. While a low score was expected the result was nonetheless surprising as it was reduced even further from the initial 0.13 in the comparison model to 0.09 in the XLS-R. Maybe an equal setup of samples would help this region, but as most data originates from the canton of Valais, the issue would probably persist. A decision has to be made in the future for the dialects in these cantons if they should either be integrated into the CA and EA region based on their proximity to them or be ignored for the time being until further data collection has been performed.

Lastly, looking at Figure 6.8 one can discern that the model was at its best during the beginning of training. This should not be the case in a normal training setup, however, as the model was trained on the SDS-200 in both the STT and classification task we assume that this is due to model overfitting on the data and not learning anything new. During the training of the XLSR-53 this was not the case as the base fine-tune model was trained on a different corpus, namely the Swiss Parliaments Corpus [63]. Thus an experiment should be made in future work which applies different corpora to demonstrate XLS-R's capabilities.

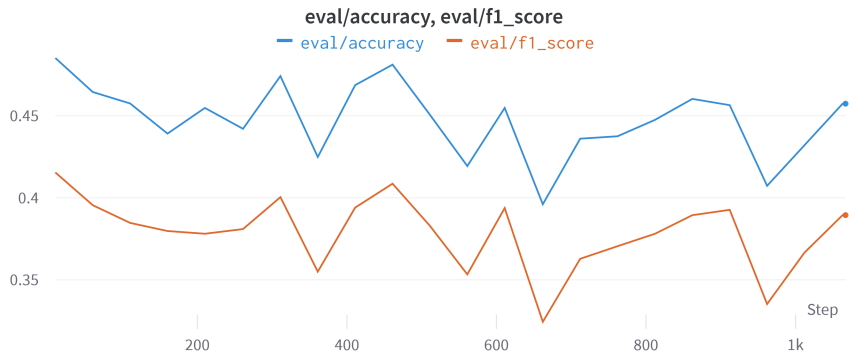


Figure 6.8: F1 and accuracy scores over time

Evidently, this still does not explain the difference in rankings of the Base and pre-train models compared to the STT experiments. A variation could be a factor in this setup as the two Full models differed by up to 2% weighted F1. However, the difference between the Base and Full models is at best 5% F1, which is too high considering the variations in the STT experiments. More research has to be conducted to give an informed decision on what the source of this contradiction is.

Region	Precision				Recall				F1-Score			
	Base-1	Full-1	Full-2	FromSTT	Base-1	Full-1	Full-2	FromSTT	Base-1	Full-1	Full-2	FromSTT
EA	0.6	**0.72**	0.67	0.70	0.62	0.46	0.54	**0.66**	0.48	0.56	0.6	**0.68**
CA	0.34	0.36	0.38	**0.39**	0.4	0.73	0.67	**0.78**	0.44	0.49	0.49	**0.52**
WA	0.41	0.51	0.49	**0.63**	0.45	**0.48**	0.5	0.4	0.43	0.5	0.49	**0.49**
HA	0.49	0.32	**0.49**	0.45	0.05	0.04	0.05	**0.05**	0.09	0.07	0.08	**0.09**

Figure 6.9: Classification results of precision, recall, and F1-score for the regions

6.3 Shared Task SwissText

All STT models that had finished training before the submission deadline on the 30th of May 2022 were sent to the shared task for evaluation. As outlined in Chapter 4, the evaluation was performed in a public and private split, with each of the splits containing 50% of the 5 hours of Graubünden dialect data. The private split was used to define the rankings of each participant's best-performing model. The results of the different STT models on the public split are displayed in Table 6.4. Our best model was the Base-1B model with a BLEU of 0.6838 and as such could not beat the baseline, which had a BLEU of 0.7044. Regrettably, the LimitedVocab models were not ready at the submission deadline, which meant that our models had a more difficult task predicting the translations. This is because the normal models had to decide between 91 characters compared to the limited models which would have only had to differentiate between 45 characters.

Model	BLEU score
CH-STT-Base-1B	0.6838
CH-STT-Base-300M-800Beam	0.6825
CH-STT-FromPretrain-1B-Full-1	0.6778
CH-STT-FromPretrain-1B-75k-800beam	0.6659
CH-STT-Base-300M	0.6724
CH-STT-FromPretrain-300M-Full-2	0.6615
CH-STT-FromPretrain-300M-75k	0.6457
CH-STT-FromPretrain-300M-Full-1	0.6371

Table 6.4: Our results on the SwissText public split

Some models in Table 6.4 have the suffix "800Beam". This refers to the beam search width during the LM evaluation, which was increased from initially 200 to 800 after consultation with our supervisors. The change increased BLEU scores by 1-2 normalized points and was thus used as the default setting for all further LM applications.

In total, apart from the organizers at FHNW and their baseline model, two participants entered the competition. The second group was part of the AXA corporation and used the same Wav2Vec2-XLS-R-1B model but without a LM to improve the translations. Rankings are displayed in Table 6.5 for the public and private split, respectively. We ranked first in both splits when discounting the baseline as a submission.

Submission	Public score	Private score
FHNW Baseline	0.7044	0.701
ZHAW Wav2Vec2	0.6838	0.681
AXA Wav2Vec2	0.5571	0.5532

Table 6.5: Ranking of participants submissions

The organizers of FHNW developed a separate Transformer based model, which was presented during the conference. Pre-training was performed using the in Chapter 4 defined datasets. After fine-tuning the model on both SDS-200 and SwissDial, they applied a KenLM model, which was heavily trained on texts used in different parliament speeches. This gave them a small advantage over the submissions, as their data matched the source of the training data. The exact implementation will be released in the SwissText 2022 proceedings paper.

6.3.1 LimitedVocab model

Since the LimitedVocab models were not ready for submission but finished before the SwissText conference was held on the 8th of June 2022, we decided to still include the models performances on the public evaluation split. Compared to the normal models, the LM-CC-100 was used instead of the LM-Wiki, as outlined in section 5.6.

Submission	BLEU score
CH-STT-Base-1B-LimitedVocab	0.7001
CH-STT-Base-300M-LimitedVocab	0.6472

Table 6.6: LimitedVocab evaluation scores public split

The 1B-LimitedVocab beat the 1B-Base from 0.6838 to 0.7001 BLEU, as shown in Table 6.6. While the model did not beat the baseline as well, reaching 70 normalized BLEU is still a large achievement for Swiss German speech translation. The increase by 1.3 BLEU compared to its normal performance on the SNF corpus was surprising however and can be grounds for discussion. A first interpretation uses the results gained during classification, which show that Wav2Vec models can recognise dialects in the eastern region of Switzerland particularly well. With the evaluation data originating from Graubünden, which is also in the east, this could mean that the model generally creates better translations for this dialect. It also shows the impact prediction vocabulary can have on a model's performance.

However, the 300M-LimitedVocab lost evaluation points compared to the 300M-Base from 0.6825 to 0.6472, which can only be explained by the model's general low performance, as illustrated in Table 6.3. Retraining a separate model with different hyperparameters would be interesting in future work to analyze the model's capabilities.

6.3.2 Learnings

Both the 300M and 1B pre-train models were beaten by the base XLS-R with the scores being noticeably lower in the pre-train models (see Table 6.4). Surprising was also that the base 300M model scored higher than the 1B pre-train models. Valuable input was provided during the discussion at the SwissText conference by the participants concerning this apparent issue. Three different approaches were discussed:

A: Apply the pre-training at the same time with the complete 436k hours of the XLS-R model

B: Better tuning of hyperparameters during fine-tuning

C: Create a monolingual Swiss German pre-trained Transformer model

Option A was discussed as a potential source for issues when training an already pre-trained model with additional data. The consensus was that the model could be prone to erroneous learnings when applying additional language data at a later time than the rest of the training data. By training a model from scratch with the Swiss German data, in addition to all corpora mentioned in the XLS-R paper [37], the model could learn the speech representations in a more generalized fashion. It was noted, however, that the computational resources required for such training are not available to any of the interested participants and as such cannot be investigated further.

The team of FHNW proposed option B, arguing that they had internally applied identical fine-tunings on the XLS-R-1B model as our CH-STT-Base-1B, but were able to achieve a BLEU of 0.72. As was the case with their baseline model, they used a LM that was applied after the initial training to reach said score.

Option C was proposed by both FHNW, as their baseline model already did these steps to a certain degree, and us, referring to the success the monolingual Wav2Vec 2.0 model had on the English language. By imitating a similar setup, we think improvements could be achieved on the translation task for Swiss German to Standard German. However, we also recognise that the required data would first have to be collected and processed. Acquiring such large quantities of data would have to be done in cooperation with media corporations that naturally have access to such data.

Chapter 7

Discussion and Outlook

Various experiments on Wav2Vec2-XLS-R have been conducted in the scope of this dissertation and as a result two systems can now be presented. The first is a STT translation system from Swiss German into Standard German. The CH-STT-Base-1B-LimitedVocab model achieves the best results with a 18.08% WER and 68.86 BLEU on the "SNF" test corpus and 68.05 BLEU on the "SDS-200" test split with a WER of 19.28%. A different model, the CH-STT-Base-1B, helped us reach first place in the SwissText "2nd Swiss German Speech to Standard German Text" shared task with a BLEU of 68.1 points on the private evaluation split. This 1B-LimitedVocab had not finished with training before the submission deadline could thus not be sent for evaluation. Later evaluation on the public split returned a BLEU of 70.01. The goal of creating a high-functioning translation system has thus been reached.

The second system is a classification model for the various Swiss German dialects, which categorised them into four distinct regions. The model achieves an overall weighted F1-score of 0.49, with the Eastern-Alemannic dialect region performing the best with a weighted F1 of 0.68. The model thus demonstrates an ability to differentiate the dialects. Previous results with the same setup could not be beaten with this system.

Analysis of the impact the additional pre-training data had on the XLS-R model show, that the effect is largely neutral or negative for the speech translation systems, but shows positive influences for classification systems. Further research is needed to remove uncertainties concerning variation and explain the apparent contradiction.

Multiple approaches are now discussed on how to increase the performance of the systems in a future work. First, the translation model could benefit from a better tuning of the hyperparameters during training. Data from FHNW illustrated that increases of up to 4 BLEU are possible. Additionally, instead of using the unlabelled data in pre-training, it could be applied in a semi- / self-supervised process and thus generate new Text-to-Speech (TTS) samples. These additional datasets could then be applied downstream for fine-tuning models and potentially increase prediction accuracy. A different approach could be the creation of a new monolingual Swiss German Transformer architecture, similar to the English base Wav2Vec2 model [16].

The data for this approach would first have to be collected, as, for reference, the base Wav2Vec2 model applied up to 60k hours of speech compared to our setup using around 2.1k hours.

Training of the classification systems has to be revisited concerning corpora and dialects used for fine-tuning. Fine-tuning a model first on a STT task has proven itself to be beneficial for both the XLSR-53 and XLS-R. However, the corpora used should differ during the classification task to avoid overfitting the model. The dialects used for training also have to be considered. Cantons that currently do not posses enough data for efficient training can hinder a models performance significantly as shown with the HA region.

Bibliography

[1] M. Śmieja, L. u. Struski, J. Tabor, B. Zieliński, and P. a. Spurek, "Processing of missing data by neural networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., May 2018.

[2] FederalStatisticalOffice, "Languages: Distribution of the National Languages," https://www.bfs.admin.ch/bfs/en/home/statistics/population/languages-religions/languages.html, [Online; Accessed 11.05.2022].

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *arXiv*, vol. abs/1810.04805, Oct. 2018.

[4] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," *CoRR*, vol. abs/1904.05862, Apr. 2019.

[5] M. Plüss, M. Hürlimann, M. Cuny, A. Stöckli, N. Kapotis, J. Hartmann, M. A. Ulasik, C. Scheller, Y. Schraner, A. Jain, J. Deriu, M. Cieliebak, and M. Vogel, "SDS-200: A Swiss German Speech to Standard German Text Corpus," *arXiv e-prints*, vol. abs/2205.09501, May 2022.

[6] SwissText, "2nd Swiss German Speech to Standard German Text Shared Task," https://www.swisstext.org/ws_swiss-german-speech-to-standard-german-text/, [Online; Accessed 06.06.2022].

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., Jun. 2017.

[8] M. Abdul-Mageed, A. Elmadany, and E. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Aug. 2021, pp. 7088–7105.

[9] A. Abdelali, H. Mubarak, Y. Samih, S. Hassan, and K. Darwish, "Arabic Dialect Identification in the Wild," *CoRR*, vol. abs/2005.06557, May 2020.

[10] A. Wu, C. Wang, J. Pino, and J. Gu, "Self-Supervised Representations Improve End-to-End Speech Translation," in *Proc. Interspeech 2020*, 2020, pp. 1491–1495.

[11] P. N. Garner, D. Imseng, and T. Meyer, "Automatic speech recognition and translation of a Swiss German dialect: Walliserdeutsch," in *Proc. Interspeech 2014*, 2014, pp. 2118–2122.

[12] M. Plüss, L. Neukom, and M. Vogel, "GermEval 2020 Task 4: Low-Resource Speech-to-Text," in *SWISSTEXT & KONVENS 2020*, ser. CEUR Workshop Proceedings, S. Ebling, D. Tuggener, M. Hürlimann, M. Cieliebak, and M. Volk, Eds., no. 2624, Held online due to COVID19, 2020.

[13] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 71–75.

[14] F. Benites, D. Tuggener, M. Hürlimann, M. Cieliebak, and M. Vogel, Eds., *Proceedings of the Swiss Text Analytics Conference 2021, Winterthur, Switzerland, June 14-16, 2021 (held online due to COVID19 pandemic)*, ser. CEUR Workshop Proceedings, vol. 2957. CEUR-WS.org, 2021.

[15] MicrosoftAzure, "General availability: Announcing the release of Swiss German dialect speech recognition support," Microsoft Azure https://azure.microsoft.com/en-us/updates/release-of-swiss-german-dialect-speech-to-text/, [Online; Accessed 16.06.2022].

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[17] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2426–2430.

[18] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: BERT for Finnish," *CoRR*, vol. abs/1912.07076, Dec. 2019.

[19] M. Hämäläinen, K. Alnajjar, N. Partanen, and J. Rueter, "Finnish Dialect Identification: The Effect of Audio and Text," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2021, pp. 8777–8783.

[20] F. Benites, P. von Däniken, and M. Cieliebak, "TwistBytes - identification of cuneiform languages and German dialects at VarDial 2019," in *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2019, pp. 194–201.

[21] M. Zampieri, S. Malmasi, Y. Scherrer, T. Samardžić, F. Tyers, M. Silfverberg, N. Klyueva, T.-L. Pan, C.-R. Huang, R. T. Ionescu, A. M. Butnaru, and T. Jauhiainen, "A report on the third VarDial evaluation campaign," in *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2019, pp. 1–16.

[22] Y. Kaufmann, "wav2vec2-large-xlsr-53-swiss-german on Swiss German fine-tuned model," HuggingFace https://huggingface.co/Yves/wav2vec2-large-xlsr-53-swiss-german, [Online; Accessed 12.12.2021].

[23] B.-H. Juang and L. Rabiner, "Speech Recognition, Automatic: History," in *Encyclopedia of Language & Linguistics (Second Edition)*, second edition ed., K. Brown, Ed. Oxford: Elsevier, 2006, pp. 806–819.

[24] A. Berdasco, G. López, I. Díaz-Oreiro, L. Quesada, and L. A. Guerrero, "User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana," in *13th International Conference on Ubiquitous Computing and Ambient Intelligence, UCAmI 2019, Toledo, Spain, December 2-5, 2019*, ser. MDPI Proceedings, J. Bravo and I. González, Eds., vol. 31. MDPI, 2019, p. 51.

[25] S. Nakamura, "Overcoming the Language Barrier with Speech Translation Technology," *Science & Technology Trends - Quarterly Review*, vol. 31, Apr. 2009.

[26] P. Ivić and D. Crystal, "Dialect," Encyclopedia Britannica https://www.britannica.com/topic/dialect, [Online; Accessed 03.05.2022].

[27] A. Etman and A. A. L. Beex, "Language and Dialect Identification: A survey," in *2015 SAI Intelligent Systems Conference (IntelliSys)*, 2015, pp. 220–231.

[28] P. von Platen, "Transformers-based Encoder-Decoder Models," https://huggingface.co/blog/encoder-decoder, [Online; Accessed 03.05.2022].

[29] R. Kulshrestha, "Transformers," https://towardsdatascience.com/transformers-89034557de14, [Online; Accessed 03.05.2022].

[30] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., May 2015.

[31] A. Galassi, M. Lippi, and P. Torroni, "Attention in Natural Language Processing," *CoRR*, vol. abs/1902.02181, Feb. 2019.

[32] R. Futrzynski, "Getting meaning from text: Self-Attention step-by-step video," https://peltarion.com/blog/data-science/self-attention-video, [Online; Accessed 11.05.2022].

[33] R. Karim, "Illustrated: Self-Attention: A step-by-step guide to self-attention with illustrations and code," https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a, [Online; Accessed 11.05.2022].

[34] R. Krüger, "Die Transformer-Architektur für Systeme zur neuronalen maschinellen Übersetzung – eine popularisierende Darstellung," *Trans-kom*, vol. 14, p. 278–324, Dec 2021.

[35] S. Rush, A. Huang, S. Subramanian, J. Sum, K. Almubarak, and S. Biderman, "The Annotated Transformer," https://nlp.seas.harvard.edu/annotated-transformer/, 2022, [Online; Accessed 18.05.2022].

[36] J. Korstanje, "Machine Learning on Sound and Audio data," https://towardsdatascience.com/machine-learning-on-sound-and-audio-data-3ae03bcf5095, [Online; Accessed 14.06.2022].

[37] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," *CoRR*, vol. abs/2111.09296, Nov. 2021.

[38] Encyclopedia-Britannica, "Phoneme," https://www.britannica.com/topic/phoneme, [Online; Accessed 01.06.2022].

[39] W. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," *CoRR*, vol. abs/1704.04222, Sep. 2017.

[40] L. Sus, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," https://neurosys.com/wav2vec-2-0-framework/, [Online; Accessed 01.06.2022].

[41] P. Le-Khac, G. Healy, and A. Smeaton, "Contrastive representation learning: A framework and review," *CoRR*, vol. abs/2010.05113, Oct. 2020.

[42] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1058–1067.

[43] P. von Platen, "Wav2Vec2 Speech Pre-Training," https://huggingface.co/blog/wav2vec2-with-ngram, [Online; Accessed 06.06.2022].

[44] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376.

[45] A. Hannun, "Sequence Modeling with CTC," https://distill.pub/2017/ctc, 2017, [Online; Accessed 15.06.2022].

[46] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 993–1003.

[47] D. Jurafsky and J. H. Martin, "N-gram Language Models," Stanford https://web.stanford.edu/~jurafsky/slp3/3.pdf, [Online; Accessed 16.06.2022].

[48] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, Jul. 2011, pp. 187–197.

[49] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318.

[50] G. LCC, "Evaluation: Understanding the BLEU Score," https://cloud.google.com/translate/automl/docs/evaluate#bleu, [Online; Accessed 03.06.2022].

[51] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," *CoRR*, vol. abs/2010.16061, Oct. 2020.

[52] SRF, "Schweizer Radio und Fernsehen," https://www.srf.ch, [Online; Accessed 06.06.2022].

[53] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common Voice: A Massively-Multilingual Speech Corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222.

[54] P. Dogan-Schönberger, J. Mäder, and T. Hofmann, "SwissDial: Parallel Multidialectal Corpus of Spoken Swiss German," *CoRR*, vol. abs/2103.11401, Mar. 2021.

[55] A. Collette, *Python and HDF5*. O'Reilly, 2013.

[56] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[57] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451.

[58] G. Wenzek, M. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, "Ccnet: Extracting high quality monolingual datasets from web crawl data," *CoRR*, vol. abs/1911.00359, Nov. 2019.

[59] Weights&Biases, "Developer tools for ML," https://wandb.ai/site, [Online; Accessed 01.06.2022].

[60] P. von Platen, "Wav2Vec2 Speech Pre-Training," https://github.com/huggingface/transformers/tree/main/examples/pytorch/speech-pretraining, [Online; Accessed 06.06.2022].

[61] R. Hotzenköcherle, *Die Sprachlandschaften der deutschen Schweiz*, ser. Reihe Sprachlandschaft Bd. 1. Aarau: Sauerländer, 1984.

[62] Dbachmann, "Brunig-napf-reuss-linie," https://commons.wikimedia.org/wiki/File:Brunig-Napf-Reuss-Linie.png, [Online; Accessed 12.06.2022].

[63] M. Plüss, L. Neukom, and M. Vogel, "Swiss Parliaments Corpus, an Automatically Aligned Swiss German Speech to Standard German Text Corpus," in *Swiss Text Analytics Conference 2021*, ser. CEUR Workshop Proceedings, F. Benites, D. Tuggener, M. Hürlimann, M. Cieliebak, and M. Vogel, Eds., no. 2957, Held online due to COVID19, 2021, pp. 1–5.

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 7

ASR Automatic Speech Recognition. 7, 11, 30

BLEU BiLingual Evaluation Understudy. 8, 9, 25–27, 34, 38, 46–48, 53–56, 66

BOS Begin-Of-Sentence. 13, 14

CA Central-High-Alemannic. 43, 44, 50, 51

CNN Convolutional Neural Network. 19, 22

CTC Connectionist Temporal Classification. 21, 22, 25

DID Dialect Identification. 8, 11, 12

DL Deep Learning. 7

EA Eastern-High-Alemannic. 43, 50, 51

EOS End-Of-Sentence. 13, 14

FHNW University of Applied Sciences and Art Northwestern Switzerland. 44, 48, 53–56

HA Highest-Alemannic. 43, 44, 50, 51, 57

LM Language Model. 22, 25, 38, 46, 47, 53, 55

LSTM Long Short-Term Memory. 9

MSA Modern Standard Arabic. 8

NLP Natural Language Processing. 7, 8, 11, 12, 25, 39

RNN Recurrent Neural Network. 12, 13

SOTA State-Of-The-Art. 8, 12, 18

SRF Schweizer Radio und Fernsehen. 31, 32

STS Speech-to-Speech. 11

STT Speech-to-Text. 4, 7, 8, 11, 25, 26, 35, 36, 39–42, 44, 45, 48–53, 56, 57

TTS Text-to-Speech. 56

WA Western-High-Alemannic. 43, 50, 51

WER Word Error Rate. 8, 25, 26, 38, 46, 47, 56

XLS-R Wav2Vec2-XLS-R. 7, 19, 23, 24, 35, 37, 39, 40, 45, 46, 48, 50, 51, 53–57, 64

XLSR-53 Wav2Vec2-XLSR-53. 9, 19, 24, 37, 45, 48–51, 57

ZHAW Zurich University of Applied Sciences. 9, 35

Appendix A

Experiment Details

Canton	Abbreviation
Aargau	AG
Appenzell Innerrhoden	AI
Appenzell Ausserrhoden	AR
Bern	BE
Basel Landschaft	BL
Basel Stadt	BS
Fribourg	FR
Glarus	GL
Graubünden / Grisons	AG
Jura	JU
Luzern	LU
Nidwalden	NW
Obwalden	OW
Sankt Gallen / Saint Gallen	SG
Schaffhausen	SH
Solothurn	SO
Schwyz	SZ
Thurgau	TG
Uri	UR
Wallis / Valais	VS
Zug	ZG
Zürich	ZH

Table A.1: Swiss German canton abbreviations

Parameter	Value
Learning rate	$3e^{-5}$
Training epochs	200
Training batch size	8
Gradient Accumulation steps	16
Warmup steps	200
Save steps	1000
Evaluation steps	500

Table A.2: Pre-Training training parameters

Parameter	Value
Learning rate	$3e^{-5}$
Training epochs	25
Training batch size	4
Gradient Accumulation steps	8
Warmup steps	100
Save steps	2000
Evaluation steps	1000

Table A.3: STT training parameters

Parameter	Value
Learning rate	$3e^{-5}$
Training epochs	25
Training batch size	4
Gradient Accumulation steps	8
Warmup steps	200
Save steps	1000
Evaluation steps	500

Table A.4: Classification training parameters

Appendix B

Code & Manual

The developed code and accompanying manual for this is thesis is available on github.zhaw.ch: https://github.zhaw.ch/Swiss-German-Dialects-Recognition/speech_translation