LE MINH NGUYEN

IMPROVING LUXEMBOURGISH SPEECH RECOGNITION WITH CROSS-LINGUAL SPEECH REPRESENTATIONS

THESIS PROJECT

UNIVERSITY OF GRONINGEN
MSC. VOICE TECHNOLOGY

Le Minh Nguyen
l.m.nguyen@student.rug.nl

Supervisor: Dr. Shekhar Nayak
s.nayak@rug.nl

Second Reader: Dr. Matt Coler
m.coler@rug.nl

university of groningen
msc. voice technology

July 2022

Contents

List of Figures

List of Tables

Dedicated to my siblings.

Acknowledgment

Enabling Speech Recognition for Luxembourgish would not have been possible without the mentorship of Dr Vladimir Despotovic. I would like to extend my deepest gratitude to Dr Vladimir Despotovic, who has supported my career goals and guided me throughout this journey.

I would like to express my deepest thanks to the whole team at the Zenter fir d'Lëtzebuerger Sprooch for validating and correcting transcriptions of audio segments. It was a pleasure to work in their interdisciplinary environment and to have fruitful discussions about standardizations of the audio labels. I am incredibly grateful to the director of the Zenter fir d'Lëtzebuerger Sprooch and my internship supervisor, Luc Marteling and Dr Sven Collette, for their guidance on this project and for having me as an intern. The internship provided me with crucial insights into the professional environment at the Luxembourgish Ministry of Education and significant networking opportunities. I must also thank the Zenter fir d'Lëtzebuerger Sprooch for providing me with computational and text data resources used during my experiments.

I would like to give my special regards to Tom Weber, Digital Director at Radio Telé Lëtzebuerg, for providing access to data resources from their website to enable speech technologies for the Luxembourgish language.

I would like to thank the HPC facilities of the University of Luxembourg for their robust infrastructure that made pre-training on massive audio files possible. Additionally, I am very grateful to my aunt, Hoa Nguyen, for providing me with her laptop, which was used to fine-tune the model checkpoints.

I also appreciate the Voice Tech Team for allowing me to complete my internship during my thesis project. Additionally, I would like to thank Dr Shekhar Nayak and Dr Matt Coler for the thesis supervision and their feedback on the report.

Finally, I thank my family and friends for their support. Especially, I am very grateful to Marie Damian for her unconditional support throughout the entire lifespan of this project.

Abstract

Luxembourgish is a West Germanic language spoken by roughly 390,000 people, mainly in Luxembourg. It remains one of Europe's under-described and under-resourced languages, not extensively investigated in the context of speech recognition. We explore the self-supervised multilingual learning of Luxembourgish speech representations to be used for the downstream speech recognition task. This thesis project improves our previous work on Luxembourgish wav2vec 2.0 models in a monolingual and transfer learning context. Our experiments show that learning cross-lingual representations are essential for low-resourced languages such as Luxembourgish. Learning cross-lingual representations and rescoring the output transcriptions with language modelling while using only 4 hours of labelled speech achieves a word error rate of 15.1% and improves the previous best result for Luxembourgish speech recognition relatively by 33.1% and absolutely by 7.5%. Increasing the amount of labelled speech to 14 hours yields a significant performance gain resulting in a 9.3% word error rate.

Index Terms—Luxembourgish, multilingual speech recognition, language modelling, self-supervised learning, wav2vec 2.0 XLSR-53, under-resourced language

1 Introduction

Lëtzebuergesch (Luxembourgish) is a West Germanic language spoken by roughly 390,000 people, mainly in Luxembourg. It remains one of Europe's under-described and under-resourced languages, not extensively investigated in the context of speech recognition. The linguistic situation in Luxembourg makes it challenging to enable speech recognition technologies. This challenge is based on two factors. First, Luxembourgish is situated in a multilingual context, resulting in frequent code-switching and usage of loan words from German and French [1]. The second factor is that it is considered an under-resourced language since written material in Luxembourgish is scarce [1]. The reason for the sparse production of written Luxembourgish material is caused by the preference for French and German usage and English in professional environments [1]. Furthermore, only in March 2017 the law was passed to advance the standardization of the Luxembourgish language, and in July 2018, the Center for the Luxembourgish Language - *Zenter fir d'Lëtzebuerger Sprooch* (ZLS) was created to implement the measures of this law[1].

The linguistic situation in Luxembourg makes it challenging to rely on labelled resources to implement speech recognition systems since they require a large amount of transcribed speech to achieve high performance. Thus, self-supervised learning has become a paradigm for determining general data representations from unlabelled examples for downstream tasks such as speech recognition. This paradigm has shown the feasibility of speech recognition based on limited labelled data with the wav2vec 2.0 model from [2].

We applied this paradigm in our unpublished work to enable speech recognition for Luxembourgish by creating the first Luxembourgish base wav2vec 2.0 model [3]. Two experiments with this model were conducted. In one experiment, a monolingual wav2vec 2.0 model was trained from scratch on Luxembourgish speech. In the other experiment, transfer learning was applied to pre-train Luxembourgish speech representations on a wav2vec 2.0 model pre-trained on the LibriSpeech corpus. The two pre-trained mod-

[1] Adda-Decker et al., "Developments of "Lëtzebuergesch" Resources for Automatic Speech Processing and Linguistic Studies", 2008.

[1] https://portal.education.lu/zls/IWWER-EIS

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

els were fine-tuned on four hours of labelled Luxembourgish speech. In both models, no language model (LM) rescoring was applied, and the Greedy algorithm[2] was used instead to decode the output of the models. These experiments with 23.05% and 22.57% test Word Error Rates (WER), respectively, represent the baseline as a benchmark for this research. Luxembourgish is an under-resourced language not extensively investigated in speech recognition, and to the best of our knowledge, there are no publicly available transcribed speech datasets. This challenging topic is connected to our interests in enabling voice technologies for the Luxembourgish language. Therefore, this thesis project suggests improvements that extend our previous work on Luxembourgish speech recognition in [3].

In this research, we investigate the following research questions:

1 *Could pre-training cross-lingual representations improve wav2vec 2.0 models that have been pre-trained on Luxembourgish solely?*

wav2vec 2.0 [2] has shown that self-supervised learning of speech representations effectively enables speech recognition for low-resourced languages while providing little labelled data. The XLSR model [4] is a multilingual speech recognition model based on wav2vec 2.0. It pre-trains on cross-lingual speech representations. Experiments have shown that cross-lingual pre-training significantly outperforms monolingual pre-training.

2 *Could language model rescoring improve the baseline Luxembourgish speech recognition models that use the Greedy algorithm for decoding?*

In the original wav2vec 2.0 [2] paper, experiments were conducted with language model rescoring and improved the model's performance significantly.

To address our research questions, we first apply a more advanced speech label pre-processing. Additionally, we investigate if multilingual wav2vec 2.0 models improve the performance of monolingual models. Finally, we use language modelling to rescore the decoding of the model. For this experiment, we use a wav2vec 2.0 model that has been pre-trained on 53 different languages from BABEL, Common Voice and MLS [5–7]. With this model, we pre-train Luxembourgish speech representations on top of it. After pre-training the multilingual model, the corrected labels, combined with LM rescoring, are used to fine-tune the model for the speech recognition task.

Our results demonstrate that pre-training cross-lingual speech representations are essential for low-resourced languages such as Luxembourgish. Learning cross-lingual representations and rescoring the output transcriptions with language modelling yield substantially better results than our monolingual model or applying transfer

[2] Greedy algorithm is a heuristic that makes a locally optimal choice at each stage. In the Connectionist Temporal Classification context, the stages represent each time step.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

[5] Gales et al., "Speech recognition and keyword spotting for low-resource languages: Babel project research at cued", 2014.

[6] Ardila et al., "Common Voice: A Massively-Multilingual Speech Corpus", 2020.

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

learning from LibriSpeech in [3]. When using only 4 hours of labelled Luxembourgish speech, our multilingual XLSR-53 wav2vec 2.0 model with LM rescoring achieves a WER of 15.1% and improves the previous best result for Luxembourgish speech recognition relatively by 33.1% and absolutely by 7.5%. Increasing the transcribed speech dataset to 14 hours, our model sets the new best WER for Luxembourgish Speech Recognition of 9.3%. Our results achieve a 58.8% relative and 13.3% absolute improvement over the best result replicated from [3].

This thesis introduces previous work on Luxembourgish speech recognition and a review of the literature that sets the research context in chapter 2 and 3. After, we will present our research issues and define our hypotheses based on the literature review in chapter 4. Having defined our research scope and hypotheses, we present our replicable methodology for this research in chapter 6. Finally, we conduct our experiments and conclude with a discussion of our results in chapter 7.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

2 *Previous Work on Luxembourgish Speech Recognition*

Luxembourgish is an under-resourced language, and to the best of our knowledge, no public transcribed speech corpus exists to enable Luxembourgish speech recognition. In [1], a study was made on the linguistic situation in Luxembourg. They described the existence of publicly available Luxembourgish audio-transcription data pairs that are useful to enable ASR systems for Luxembourgish. Based on these findings, there have been first attempts to Luxembourgish speech recognition in [3, 20, 23]. We will study their approaches to Luxembourgish speech recognition in chronological order.

In [23], they proposed the study on acoustic similarities between Luxembourgish and its contact languages such as German, French and English. They used speech alignment and recognition to analyze the acoustic similarities between these languages. In their experiments, they created monolingual acoustic models for each contact language, a multilingual model pooled speech data from the three contact languages, and a native Luxembourgish acoustic model trained on 1200 hours of speech in an unsupervised manner. Their unsupervised acoustic model training process generates pseudo labels at each training iteration using context-independent acoustic models with a pronunciation lexicon and a language model. These generated labels are used for pseudo-supervised acoustic model training. With each iteration, more audio is recognized, and the acoustic model becomes more accurate by training on more context. One of their primary research questions investigates how unsupervised monolingual Luxembourgish acoustic models perform compared to supervised monolingual and multilingual models trained on the contact languages in a force alignment setup. Additionally, they explored how the unsupervised Luxembourgish monolingual model performs in ASR. Their unsupervised Multilayer Perceptron (MLP) achieved an ASR WER of 25.6%.

The study in [20] focuses on lightly supervised and semi-supervised learning to improve Deep Neural Network (DNN) acoustic models for Luxembourgish ASR. In their work, 17.7 hours of transcribed

[1] Adda-Decker et al., "Developments of "Lëtzebuergesch" Resources for Automatic Speech Processing and Linguistic Studies", 2008.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

[20] Veselý et al., "Lightly Supervised vs. Semi-supervised Training of Acoustic Model on Luxembourgish for Low-resource Automatic Speech Recognition", 2018.

[23] Adda-Decker, Lamel, and Adda, "Speech alignment and recognition experiments for Luxembourgish", 2014.

Luxembourgish speech from Contact Centers (CCs) were collected through the BISON[1] project. Thus, they do not rely on a fully unsupervised approach as in [23] to implement Luxembourgish speech recognition. Their research considered augmenting the training in two approaches to improve the acoustic models. In the first experiment, they applied semi-supervised training by pseudo labelling untranscribed audio collected from the CC target domain with a seed acoustic model. Then, they retrained an acoustic model using a combination of transcribed and pseudo-labelled audio. Lightly supervised training was used in the second experiment by augmenting the labelled CC training data with an out-of-domain and inexactly transcribed speech from Luxembourgish parliament sessions. Overall their data augmentation methodology improved their baseline ASR system trained only on the 17.7 hours of labelled speech. Their final WER of 34.4% is poor and does not yield an improvement over [23] for Luxembourgish speech recognition.

In our previous unpublished work [3], we trained the first Luxembourgish base wav2vec 2.0 models, which are domain-specific to recognize speech from broadcast news. For this study, we collected and validated 4 hours of labelled Luxembourgish speech from *RTL.lu*[2]. In addition, we scraped 842 hours of unlabelled Luxembourgish speech from the same domain. With the limited labelled dataset, we applied the self-supervised learning framework wav2vec 2.0 from [2] to enable speech recognition for Luxembourgish. We conducted two wav2vec 2.0 model experiments. One monolingual model pre-trained speech representations from scratch using the 842 hours of unlabelled Luxembourgish speech. In the other experiment, we applied transfer learning and fine-tuned unlabelled Luxembourgish speech on a wav2vec 2.0 model pre-trained on the LibriSpeech corpus. On top of these two pre-trained models, a randomly initialized linear layer was placed for the speech recognition task. The linear layer of the two models was fine-tuned using the Connectionist Temporal Classification (CTC) loss on 4 hours of labelled audio, which was not correctly pre-processed. The speech labels were not clean and contained many issues[3], e.g. not spelt out numbers or abbreviations, removal of apostrophe variations. Although without having access to the same evaluation dataset, this research and [23] target Luxembourgish speech recognition in a news broadcasting context. Therefore, the results of these experiments show an improvement in terms of overall WER over [23]. The models achieved a validation WER of 25.1% and 23.5%, respectively, without language modelling and using the Greedy search only when decoding the output transcriptions. The WER obtained in this study improves the previous best result for Luxembourgish speech recognition relatively by 8.2%

[1] http://bison-project.eu/

[23] Adda-Decker, Lamel, and Adda, “Speech alignment and recognition experiments for Luxembourgish”, 2014.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

[2] RTL.lu is part of the RTL Group and is a news website containing online content for the Radio Télé Lëtzebuerg (Radio TV Luxembourg), which is the principal television channel in Luxembourg broadcasting in Luxembourgish.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[3] Most of the issues relate to the non-verbalization of tokens. A detailed list of issues will be given in the methodology chapter 6.

and absolutely by 2.1%.

3 Literature Review

AFTER reviewing what has been done previously for Luxembourgish speech recognition, we perform in this chapter a literature review and synthesize what approaches have been taken in the field to enable speech recognition in a low-resource setting. In the previous study [3], we collected Luxembourgish audio resources to create a speech corpus to train wav2vec 2.0 models to recognize Luxembourgish speech. Additionally, these models were not rescored by language modelling nor trained on proper normalized speech labels. Therefore, in addition to synthesizing previous approaches to low-resource ASR, we review previous methods for creating speech corpora for speech technologies, text normalization of speech labels and language modelling in speech recognition models.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

We compile a list of keyword sequences around this central topic. The keywords guide the collection of literature. The following are the keyword sequences for each subject that we used to look up literature on *SmartCat* and *Google Scholar*:

Design of Speech Corpus: speech corpus, spoken corpus, design of speech corpus, speech recognition corpus, low-resource language speech corpus, Luxembourgish speech corpus

Text Normalization of Speech Transcriptions text normalization, text pre-processing, speech transcription normalization, speech recognition text normalization, number expansion, abbreviations expansion

Low-resource Automatic Speech Recognition low-resource speech recognition, low-resource automatic speech recognition

Language modelling in Speech recognition: speech recognition language modelling, speech recognition language model rescoring, automatic speech recognition language modelling, automatic speech recognition language model rescoring

Collection of Literature

In a first step, we use each keyword sequence on its own to collect research from the literature related to our research topic based on the criteria of how recent the literature is and if it is well cited compared to the other results in the same query. Then for each subject collection, we evaluate to select a subset of relevant literature that showcases a novel approach or is aligned with a research trend.

Design of Speech Corpus

For the *Design of Speech Corpus* subject, we found the following related literature using our defined keywords:

[6] Ardila et al., "Common Voice: A Massively-Multilingual Speech Corpus", 2020.

[8] Magueresse, Carles, and Heetderks, "Low-resource languages: A review of past work and future challenges", 2020.

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

[9] Panayotov et al., "Librispeech: an ASR corpus based on public domain audio books", 2015.

To explore the process and design principles of creating a speech corpus for ASR, we investigate the detailed methodologies in [6, 7, 9].

Text Normalization of Speech Transcriptions

For the *Text Normalization of Speech Transcriptions* subject, we found the following related literature using our defined keywords:

[10] Mansfield et al., "Neural Text Normalization with Subword Units", 2019.

[11] Zhang et al., "Neural models of text normalization for speech applications", 2019.

[12] Yolchuyeva, Németh, and Gyires-Tóth, "Text normalization with convolutional neural networks", 2018.

For this subject, we choose [10–12]. Their approaches are all aligned in studying neural methods to solve text normalization and comparing them to traditional rule-based approaches.

Low-resource Automatic Speech Recognition

For the *Low-resource Automatic Speech Recognition* subject, we found the following related literature using our defined keywords:

[13] Aldarmaki et al., “Unsupervised automatic speech recognition: A review”, 2022.

[14] Babu et al., “XLS-R: Self-supervised cross-lingual speech representation learning at scale”, 2021.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., “Unsupervised Cross-lingual Representation Learning for Speech Recognition”, 2020.

[15] Fantaye, Yu, and Hailu, “Investigation of automatic speech recognition systems via the multilingual deep neural network modeling methods for a very low-resource language, Chaha”, 2020.

[16] Hsu, Chen, and Lee, “Meta learning for end-to-end low-resource speech recognition”, 2020.

[17] Karunathilaka et al., “Low-resource sinhala speech recognition using deep learning”, 2020.

[18] Yi et al., “Applying wav2vec2.0 to speech recognition in various low-resource languages”, 2020.

[19] Srivastava et al., “Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages.”, 2018.

[21] De Wet et al., “Speech recognition for under-resourced languages: Data sharing in hidden Markov model systems”, 2017.

[22] Cui et al., “Multilingual representations for low resource speech recognition and keyword search”, 2015.

[24] Besacier et al., “Automatic speech recognition for under-resourced languages: A survey”, 2014.

[25] Thomas et al., “Deep neural network features and semi-supervised training for low resource speech recognition”, 2013.

[26] Le and Besacier, “Automatic speech recognition for under-resourced languages: application to Vietnamese language”, 2009.

Among these results, we analyse the modern approaches from [2, 4, 14] that enable speech recognition in a low-resource setting. Additionally, we review [24] to see which approaches have been taken previously before self-supervised learning became feasible for speech recognition.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

[14] Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale", 2021.

[24] Besacier et al., "Automatic speech recognition for under-resourced languages: A survey", 2014.

Language Modelling in Speech Recognition

For the *Language Modelling in Speech Recognition* subject, we found the following related literature using our defined keywords:

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[27] Xu et al., "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition", 2018.

[28] Kumar et al., "Lattice rescoring strategies for long short term memory language models in speech recognition", 2017.

[29] Chan et al., "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition", 2016.

[30] Arisoy et al., "Bidirectional recurrent neural network language models for automatic speech recognition", 2015.

[31] Kuhn and De Mori, "A cache-based natural language model for speech recognition", 1990.

We select the studies [2, 27, 28, 30] which show the trend where the research of Language Modelling in ASR is headed using neural architectures to improve traditional approaches.

Synthesis of literature

For each subject, we chronologically synthesize the selected literature from the previous section. We review the approaches that have been taken previously to enable speech recognition for low-resourced languages. Additionally, we investigate the trend of new methods for low-resource ASR and study their relation to earlier literature in the field. Furthermore, our research attempts to improve Luxembourgish speech recognition. Hence we compare the approaches applied in the field to the previous work that researched Luxembourgish ASR to determine its weak points. Finally, we suggest improvements from these shortcomings and frame them as research questions based on the theoretical foundation synthesized from the selected literature.

Design of Speech Corpus

In [1], a study was made on the linguistic situation in Luxembourg. They described the existence of Luxembourgish audio-transcription data pairs that are useful to enable ASR systems for Luxembourgish. Thus with the knowledge of this investigation, it is important to study the design principles of creating a speech corpus for a language that does not have any public speech corpora for reference.

The procedure of creating the LibriSpeech corpus was presented in [9]. The LibriSpeech corpus is a good reference point for creating our Luxembourgish corpus since in both cases, the raw data represents read speech where the audio and transcription are freely available in the public domain. The LibriSpeech corpus is obtained from audiobooks from the LibriVox project. To produce the LibriSpeech corpus, they pre-processed each audiobook's text by upcasing its text, expanding frequent abbreviations and removing punctuations. Then, the Kaldi toolkit is used to recognize the audiobooks to generate transcriptions. In a first alignment stage, they use the Smith-Waterman alignment algorithm [32] to identify the common subsequences of words among the generated transcription and the audiobook text. In most cases, the largest region of similarity corresponds to an entire chapter. Each transcription word in that region of similarity equivalent to the reference is marked with a high confidence metric. Each region of similarity is divided into shorter segments of 32 seconds or less, where the splits are done at silence intervals. In a second alignment stage, every segment is filtered out where the transcription is likely incorrect. With a subset of 32 seconds long and accurate audio segments, each segment is chunked into smaller segments before being included in the corpus. For the data selection, they collected speaker information to make sure that a speaker is not represented in different corpus splits. Additionally, they labelled each audio segment with the gender of the speaker to ensure a gender balance within the corpus splits.

The approach in [7] to create the MLS corpus is similar to the methodology from [9] although in a multilingual context. The difference is that they only have one audio segmentation stage in their data processing pipeline. They started by segmenting the audiobooks from the LibriVox domain into 10-20 seconds segments and generated pseudo labels using their in-house trained acoustic models. For the transcript retrieval process, they also used the Smith-Waterman alignment algorithm to find the best matching subsequences of words. With the alignment algorithm, for each audio segment, a candidate target label from the audiobook source text is generated that matches best with the pseudo label. Each candidate transcription

[1] Adda-Decker et al., "Developments of "Lëtzebuergesch" Resources for Automatic Speech Processing and Linguistic Studies", 2008.

[9] Panayotov et al., "Librispeech: an ASR corpus based on public domain audio books", 2015.

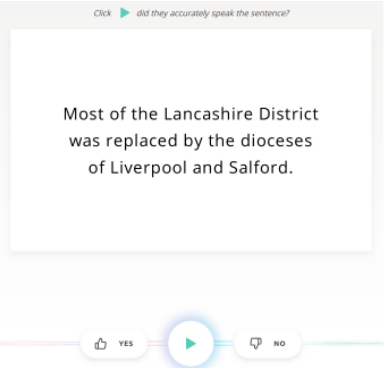
[32] Smith and Waterman, "Identification of common molecular subsequences", 1981.

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

is rejected when the WER is above 40% between the candidate and pseudo label.

The two previous studies apply subsequences alignment to match audio segments with transcriptions to create speech corpora for speech technology research. However, this design choice scales poorly to under-resourced languages [6]. In [6], they create open source tools that scale to new and potentially under-resourced languages. The open-source tools permit community members to submit text sentences for any language, record and validate voices in that language. In this crowd-sourcing approach, a corpus for a new language can be created without relying on aligning audio-transcription pairs. In order to request the creation of a new language speech corpus, text prompts have to be submitted by the community. Other community members read these text prompts and record their utterances. The read samples are verified by other contributors with a voting system. With two up-votes, the utterance becomes valid. While with two downvotes, it is invalid. This voting system is illustrated in Figure 3.1. During the corpus creation, the data is split into 80%, 10% and 10% for the train, dev and test sets, respectively. The dataset splits were done considering keeping one speaker's recoding only in one of the splits.

[6] Ardila et al., "Common Voice: A Massively-Multilingual Speech Corpus", 2020.



Click ▶ did they accurately speak the sentence?

Most of the Lancashire District was replaced by the dioceses of Liverpool and Salford.

Figure 3.1: The Common Voice voting system to verify the validity of read samples of submitted text prompts [6].

The trend in creating training data for speech technology research in high-resource languages is to use alignment algorithms to segment long audio files according to text transcriptions. However, this approach does not scale well for under-resourced languages. In many under-resourced languages, no existing speech recognition models are available to apply this alignment technique of matching recognized transcription with the target text. Additionally, labelled data is scarce, and we cannot allow rejecting every incorrectly transcribed utterance that could become useful as training data. Furthermore, there is no public implementation of the Smith-Waterman alignment algorithm for matching audio-transcription pairs. On the other hand, the crowd-sourcing approach in [6] is scalable for Luxembourgish. However, it is not ideal for our research either since, at the time of writing this thesis, the Common Voice web application UI has not been localized yet for Luxembourgish, and only seven sentences out of 5000 were submitted by community contributors[1]. Relying on the Luxembourgish community to create an open-source speech corpus would take too long and is not feasible for our research. In both approaches, to create a fair evaluation of speaker generalization, they ensured that a speaker's utterances would only appear in one dataset split and that the splits are gender-balanced with the additional meta-information made accessible from the raw data source. This is not possible for our speech data. In [3], we scraped only the

[1] As of 18th June 2022.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

audio-transcription pairs without any additional Personal Identifiable Information (PII) such as names or gender information to ensure that our automatic data scraping was compliant with the General Data Protection Regulation (GDPR) [33].

[33] *2018 reform of EU data protection rules*, 2018

Text Normalization of Speech Transcriptions

The two resources from [7, 9] on creating speech corpora for speech technology research, presented also their methodology on normalizing the transcription text. For the LibriSpeech corpus, each text of an audiobook was converted to uppercase. Special characters such as punctuations were removed, and abbreviations were expanded. To create the MLS corpus, they normalized the text first by removing special characters such as punctuations. They joined words together that were separated by end-of-line hyphenation. Characters outside the valid Unicode characters of a language were filtered out. After creating the true target label for an audio segment, they applied a post text processing based on heuristics which does not represent the ideal solution. In this processing pipeline, they replaced the numbers in the matched text with the aligned words from the pseudo label. They chose this solution since number-to-words conversion libraries like *num2words*[2] fail in situations where the conversion depends on the context. For example, a number 2020 can be pronounced as *two thousand and twenty*, but *twenty twenty* would also be correct. Furthermore, they use rule-based substitutions to deal with hyphens and apostrophes.

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

[9] Panayotov et al., "Librispeech: an ASR corpus based on public domain audio books", 2015.

[2] https://pypi.org/project/num2words/

We notice a trend for text normalization in recent literature such as [10–12]. They study the neural approach to solve text normalization and believe that these neural models relieve the burden of creating grammars represented by Finite State Transducers (FSTs). FSTs are finite state machines that map between two sets of strings and are frequently used in past approaches to building grammars that handle text normalization. [10–12] share the idea that text normalization is the process of verbalizing Non-Standard Words (NSWs) or also called semiotic classes following Taylor in [34]. Semiotic class instances denote numbers, monetary amounts, times, dates, etc. [12] proposed a CNN model and [10, 11] presented their sequence-to-sequence models to deal with text normalization without relying on linguistic knowledge to define hand-written language-specific grammars. The proposed method in [12] identifies first the class of each token in a sentence. Then it generates the verbalization of a token according to its class within a sentence context. Whereas, in [10, 11], they treat text normalization as a sequence-to-sequence problem and model the sentential context to compute the verbalization of a token in a sequence

[10] Mansfield et al., "Neural Text Normalization with Subword Units", 2019.

[11] Zhang et al., "Neural models of text normalization for speech applications", 2019.

[12] Yolchuyeva, Németh, and Gyires-Tóth, "Text normalization with convolutional neural networks", 2018.

[34] Taylor, *Text-to-speech synthesis*, 2009.

of words without relying on a POS tagger or grammar. Even though [11] described that modelling text normalization as a machine translation problem performs very well overall, with neural translation, sometimes inappropriate verbalizations are predicted, such as *'3 cm'* are verbalized as *'three kilometres'*. Thus, it is still important to define grammars with FSTs to supervise the neural approaches.

[11] Zhang et al., "Neural models of text normalization for speech applications", 2019.

Using machine translation to model text normalization as a sequence-to-sequence problem works well for high-resource languages. However, it is not feasible to apply neural approaches to model text normalization for low-resourced languages since no input-label pairs exist that represent the verbalizations of semiotic class instances.

Low-resource Automatic Speech Recognition

[24] state that in the past, a standard speech recognition system was based on statistical modelling. In general, these ASR systems used stochastic HMM-based approaches that consisted of three main components; acoustic model, pronunciation dictionary and language model. These models require many labelled data for the development of ASR systems. However, transcribed speech for under-resourced languages is scarce. Thus, the essential task in low-resource speech recognition research is the collection of data and creating a labelled speech corpus for an under-resourced language. In many under-resourced languages, speech from broadcast news and parliamentary sessions present a good starting point. Many researchers rely on manually transcribing the available audio recordings for supervised learning. Obtaining audio transcriptions for an under-resourced language is generally a complex task. Thus, other researchers apply unsupervised or lightly-supervised learning to train acoustic models to reduce the burden of finding appropriate language experts to transcribe the audio collection. Lexical modelling used grapheme-based approaches where each word in the pronunciation lexicon is decomposed into its basic acoustic units, graphemes, represented by the acoustic model. The language model component reestimates the probability of the output word sequence emitted by the speech decoder. Language models in the past of stochastic HMM-based approaches were commonly based on n-grams. n-gram language models approximate the maximum likelihood of a word sequence based on a reference text corpus. Therefore, statistical language modelling requires extensive text training data. However, access to large text corpora is challenging in the context of under-resourced languages. [35] proposed a solution to text data sparseness with a word decomposition algorithm that reduces a high out-of-vocabulary rate and improves the lack of extensive training text data required in statistical

[24] Besacier et al., "Automatic speech recognition for under-resourced languages: A survey", 2014.

[35] Pellegrini and Lamel, "Are audio or textual training data more important for ASR in less-represented languages?", 2008.

language modelling.

We observe a new trend in [2, 4, 14] to approach low-resource speech recognition. They are based on self-supervised learning of speech representations and fine-tuning on only a few minutes of labelled speech, and this paradigm outperforms previous semi-supervised methods. Additionally, [4, 14] pre-train a single wav2vec 2.0 model from [2] with speech representations from multiple languages and showcase that pre-training cross-lingual speech representations outperform significantly monolingual acoustic models. Therefore, [4, 14] released their pre-trained multilingual acoustic model checkpoints to advance research in low-resource speech recognition. [4] released their XLSR-53 model pre-trained on 53 languages from the BABEL, CommonVoice and MLS, representing 56 thousand hours of speech audio. [14] released their XLS-R model pre-trained on 436 thousand hours of speech from 128 different languages.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

[14] Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale", 2021.

Language Modelling in Speech Recognition

As previously mentioned, the language model component of a speech recognition system reestimates the probability of the output word sequence emitted by the speech decoder. Language models in the past of stochastic HMM-based approaches were commonly based on n-grams [24]. n-gram language models approximate the maximum likelihood of a word sequence based on a reference text corpus.

[24] Besacier et al., "Automatic speech recognition for under-resourced languages: A survey", 2014.

However, we observe a trend in recent studies from [27, 28, 30] where Recurrent Neural Network Language Models (RNNLMs) are used to reestimate decoded speech recognition outputs. RNNLMs have been shown to outperform traditional n-gram models since RNNs model longer distance contextual information from past inputs than n-gram models [30]. In [30], they proposed bidirectional RNNs and Long Short Term Memory (LSTM) neural networks for language modelling in speech recognition. They proposed the bidirectional architecture since previous unidirectional RNNLMs only predict outputs from past inputs while bidirectional recurrent LMs also condition on future inputs. They found that bidirectional RNNs are significantly outperforming unidirectional RNNs while bidirectional LSTMs do not present any improvements. Although RNNLMs outperform traditional n-gram LMs on speech recognition tasks, the RNN approaches are more computationally expensive than their n-gram counterpart for decoding. In [27, 28], they evaluate and propose lattice-rescoring algorithms to take advantage of RNNLMs in speech recognition systems efficiently. Lattice-rescoring generates a word-lattice from a first decoding forward pass, and an RNNLM

[27] Xu et al., "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition", 2018.

[28] Kumar et al., "Lattice rescoring strategies for long short term memory language models in speech recognition", 2017.

[30] Arisoy et al., "Bidirectional recurrent neural network language models for automatic speech recognition", 2015.

rescores the lattice. n-gram approximation is often applied to reduce the search space when rescoring the lattice.

In the official wav2vec 2.0 paper [2], they applied a Transfomer-based LM to reestimate the speech decoding. Table 9 of Appendix C in their paper shows that Transfomer-based LM yields better results than n-gram language modelling for rescoring. However, the performance gain is less significant for the difference between n-gram and Transfomer-based language modelling than the difference between n-gram LM and no language model rescoring [36]. Based on Table 9 of Appendix C in [2], [36] elaborates that rescoring wav2vec 2.0 fine-tuned on only 10 minutes of labelled data with an n-gram reduces the WER relatively by around 80% while the Transfomer-based LM only generates a relative 23% WER improvement over the n-gram rescoring. For a large wav2vec 2.0 checkpoint that was pre-trained on a high amount of speech data, a Transfomer-based LM improves the WER relatively by only 8% in comparison to n-gram language modelling. In contrast, n-gram rescoring produces a relative WER reduction of 21% compared to not using LM resoring at all. Additionally, [36] explains that Transfomer-based LM rescoring is very computationally expensive since it requires a complete forward pass to estimate the likelihood of the following word. Compared to modern Transfomer-based language models, n-gram LMs are very fast and computationally expensive as queries in a look-up table. Therefore the usage of n-gram LM is favoured over Transformer-based LM since n-gram has notably a little computational cost.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[36] Platen, *Boosting Wav2Vec2 with n-grams in HF Transformers*, 2022.

We notice a similar trend in [27, 28, 30] that even though language modelling based on neural network architectures are outperforming their n-gram counterpart, neural-based LMs are still very computationally expensive. Further, to successfully train a neural-based LM that performs well, we need to provide it with large text resources, which is hard to obtain for under-resourced languages.

[27] Xu et al., "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition", 2018.

[28] Kumar et al., "Lattice rescoring strategies for long short term memory language models in speech recognition", 2017.

[30] Arisoy et al., "Bidirectional recurrent neural network language models for automatic speech recognition", 2015.

Summary of Key Findings and Definition of Research Scope

In this summary, we raise the critical findings of the literature review and define the research scope of our thesis project.

In recent studies, alignment algorithms such as the Smith-Waterman common subsequences identification algorithm are often used to segment long audio files according to text transcriptions to create training data for speech technology research in high-resource languages. However, this approach does not scale well for under-resourced languages. In many under-resourced languages, no speech recognition models are available to generate audio transcription and align it with the target text. Additionally, labelled data is scarce in an under-resourced context, and we cannot allow rejecting every incorrectly transcribed utterance that could become useful as training data. The lack of an open-source segmentation library for this algorithm makes it even harder to consider this segmentation approach for creating a speech corpus for an under-resourced language such as Luxembourgish, for which we already have an acoustic model. On the other hand, the crowd-sourcing approach in [6] is scalable for Luxembourgish. This approach is not ideal for our research either since the Common Voice web application UI has not been localized yet for Luxembourgish, and not enough text prompts have been submitted yet to record Luxembourgish utterances. Relying on a community-backed speech corpus would take too long and is not feasible for our research. In both approaches, they ensured that a speaker's utterances would only appear in one dataset split and that the splits were gender-balanced to create a fair evaluation of speaker generalization. This is impossible for our speech data since we do not have any meta-information about the speakers to ensure speaker generalization. Considering that we are not building a general ASR system but only recognizing Luxembourgish speech from broadcast news, we could limit our scope not to enforce speaker generalization. As an approach to design our speech corpus, we could take a combination of the audio-text alignment procedure from [7, 9] and the crowdsourced data validation from [6]. However, with labelled Luxembourgish speech being scarce, we cannot filter out almost correct transcribed audio segments, which could become useful training data. Therefore, for this study, we implement the idea of the human in the data processing pipeline from [6]. We design a tool where a user can validate aligned audio-transcription pairs and correct the transcription if necessary.

[10–12] share the idea that text normalization is the process of verbalizing NSWs. NSWs denote numbers, monetary amounts, times, dates, etc. In past approaches, Finite State Transducers (FSTs) are used to build grammars that handle text normalization. We notice a trend for text normalization in recent literature [10–12]. They all study the neural approach to solve text normalization and believe that these neural models relieve the burden of relying on linguistic knowledge to define hand-written language-specific grammars represented by FSTs. In neural approaches, text normalization is modelled as a sequence-to-sequence problem which works well for high-resourced languages. However, for our speech corpus, it is not feasible to apply neural approaches to model text normalization for Luxembourgish text since there exist no Luxembourgish input-label pairs that represent the verbalizations of semiotic class instances. Therefore, we rely on our Luxembourgish linguistic knowledge to

[6] Ardila et al., "Common Voice: A Massively-Multilingual Speech Corpus", 2020.

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

[9] Panayotov et al., "Librispeech: an ASR corpus based on public domain audio books", 2015.

[10] Mansfield et al., "Neural Text Normalization with Subword Units", 2019.

[11] Zhang et al., "Neural models of text normalization for speech applications", 2019.

[12] Yolchuyeva, Németh, and Gyires-Tóth, "Text normalization with convolutional neural networks", 2018.

craft language-specific grammar in our text processing procedure.

For training acoustic models, self-supervised learning of speech representation is still widely used in recent high and low-resource speech recognition research. Thus, we continue to experiment with this approach. We investigate the learning of cross-lingual speech representations to verify the claim of [4, 14] that learning of cross-lingual speech representations significantly outperform monolingual wav2vec 2.0 models pre-trained only on a single language.

We rely on n-gram LMs to reestimate our speech decodings since we noticed a common trend in [27, 28, 30, 36] that even though language modelling based on neural network architectures are outperforming their n-gram counterpart, neural-based LMs are still very computationally expensive. Additionally, to successfully train a neural-based LM that performs well, we need to provide it with large text resources, which is hard to obtain for the Luxembourgish language. Furthermore, n-gram LM rescoring is well integrated into the Hugging Face wav2vec 2.0 implementation presented in [36].

The research goals of this thesis aim to improve our previous work on Luxembourgish ASR. In our previous work, we trained a monolingual model in one experiment and used transfer learning in another to enable speech recognition for Luxembourgish. These models were neither rescored by language modelling nor trained on properly normalized speech labels. Therefore, we suggest improvements based on the findings of the literature review to the shortcomings of the previous study. We apply a distinct methodology using multilingual wav2vec 2.0 XLSR models that pre-train cross-lingual speech representations to improve speech recognition performance. Additionally, we increase the size of our speech corpus and implement a more advanced pre-processing pipeline to normalize our collected Luxembourgish speech labels. Further, we use a language model to rescore the decoding of our ASR models. We define the following research questions to frame this thesis within the current state of the art based on the findings from the literature review:

1 *Could pre-training cross-lingual representations improve wav2vec 2.0 models that have been pre-trained on Luxembourgish solely?*

   wav2vec 2.0 [2] has shown that self-supervised learning of speech representations effectively enables speech recognition for low-resourced languages while utilizing little labelled data. The XLSR model [4] is a multilingual speech recognition model based on wav2vec 2.0. It pre-trains on cross-lingual speech representations. Experiments have shown that cross-lingual pre-training significantly outperforms monolingual pre-training.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

[14] Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale", 2021.

[27] Xu et al., "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition", 2018.

[28] Kumar et al., "Lattice rescoring strategies for long short term memory language models in speech recognition", 2017.

[30] Arisoy et al., "Bidirectional recurrent neural network language models for automatic speech recognition", 2015.

[36] Platen, *Boosting Wav2Vec2 with n-grams in HF Transformers*, 2022.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

2 *Could language model rescoring improve the baseline Luxembourgish speech recognition models that use the Greedy algorithm for decoding?*

In the original wav2vec 2.0 [2] paper, experiments were conducted with language model rescoring and improved the model's performance significantly.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

4 Research Questions and Hypotheses

After reviewing the literature, we define our research questions and hypotheses for this thesis to suggest improvements over our previous work on self-supervised Luxembourgish ASR from [3]. The research questions and hypotheses are framed within the research scope defined by our findings in the literature review.

The first research question explores if pre-training cross-lingual representations could improve wav2vec 2.0 models that have been pre-trained on Luxembourgish solely. [2] has shown that self-supervised learning of speech representations effectively enables speech recognition for low-resourced languages while providing little labelled data. The wav2vec 2.0 XLSR model [4] is a multilingual speech recognition model that pre-trains on cross-lingual speech representations. Experiments have shown that cross-lingual pre-training significantly outperforms monolingual pre-training. Therefore, we hypothesize, following [4] that pre-training cross-lingual representations will improve monolingual wav2vec 2.0 models that have been trained solely on Luxembourgish. If this hypothesis is invalidated, that would suggest that Luxembourgish wav2vec 2.0 models do not benefit from cross-lingual pre-training. This will call into question the claim from [4] that pre-training cross-lingual representations improve the ASR performance for low-resourced languages.

The second research question investigates if LM rescoring could improve the baseline Luxembourgish speech recognition models that use the Greedy algorithm for decoding. Experiments were conducted with LM rescoring in the original wav2vec 2.0 [2] paper and significantly improved the model's performance. Following [24], we hypothesize that using LMs for rescoring will further improve the Luxembourgish wav2vec 2.0 models. If this hypothesis is invalidated, that would suggest that Luxembourgish speech recognition models do not benefit from language model rescoring. This will call into question the claim from [24] that incorporating a language model to rescore the speech decoder improves the recognition accuracy by emitting the best recognition hypothesis.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

[24] Besacier et al., "Automatic speech recognition for under-resourced languages: A survey", 2014.

5 Background

We investigate the models and methodologies in [2] and [4] that proposed the novel self-supervised speech recognition framework wav2vec 2.0 and wav2vec 2.0 XLSR-53 respectively.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

Self-supervised learning

Self-supervised learning is considered a subset of unsupervised learning. In contrast to supervised learning, this paradigm does not rely on labelled data. It produces pseudo labels for the supervision task and learns general representations used for downstream tasks such as speech recognition. The self-supervised paradigm is commonly used in Generative Adversarial Networks (GANs) and contrastive learning [39]. We will focus on the latter use case, applied in wav2vec 2.0 to group similar learned representations.

[39] Yolyan, *Review on Self-Supervised Contrastive Learning*, 2021.

wav2vec 2.0

The wav2vec 2.0 framework was proposed by [2] and is based on self-supervised learning. This framework is situated between supervised and unsupervised learning and is used to train a model in two steps. First, the wav2vec 2.0 model pre-trains quantized speech representations to initialize the network weights of the acoustic model. In the second, the model is fine-tuned with limited supervision. An illustration of this model is given in Figure 5.1.

First, we will describe all the modules of the wav2vec 2.0 model and define the objective that is learned during the pre-training step:

1. **Feature encoder.** The first component of the model is the feature encoder which represents a multi-layer temporal convolutional encoder:

$$f : \mathcal{X} \mapsto \mathcal{Z} \tag{5.1}$$

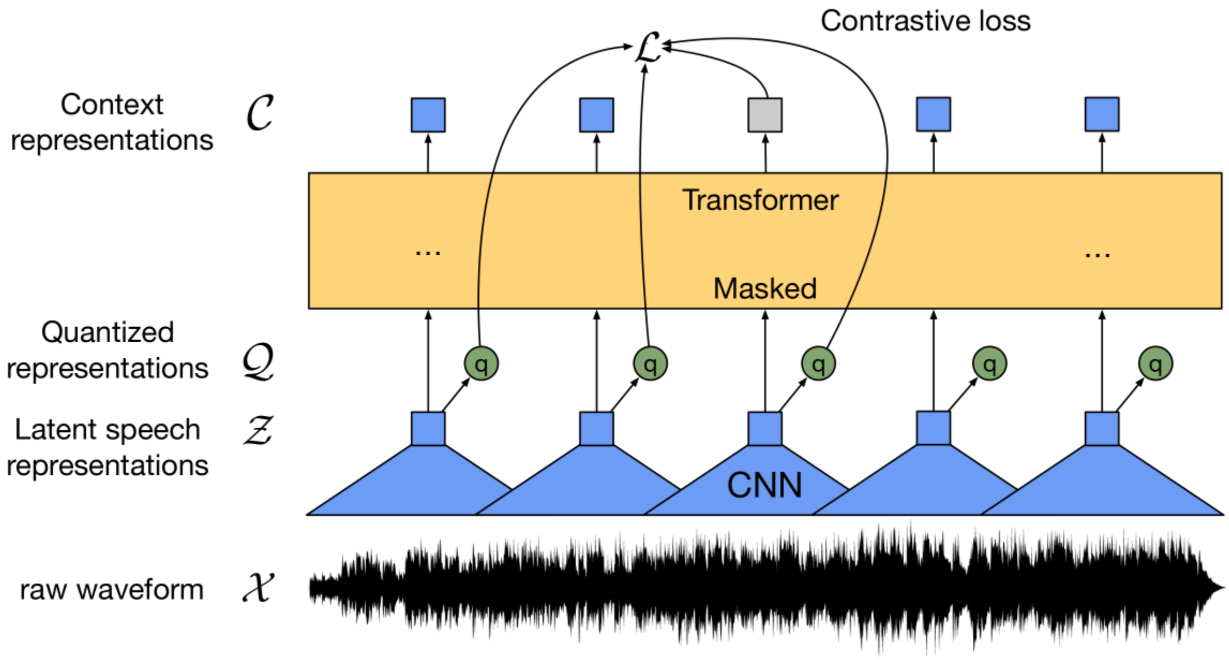

Contrastive loss
$\mathcal{L}$
Context representations $\mathcal{C}$
Transformer
...
Masked
...
Quantized representations $\mathcal{Q}$
q
q
q
q
q
Latent speech representations $\mathcal{Z}$
CNN
raw waveform $\mathcal{X}$

Figure 5.1: An Illustration of the wav2vec 2.0 framework. This framework learns jointly contextualized and discrete quantized speech representations to solve a contrastive task [2].

where $\mathcal{X}$ is a raw audio input and $\mathcal{Z}$ is the output containing latent speech representations $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ for T time-steps. The output of the feature encoder is normalized before being activated by a Gaussian Error Linear Unit (GELU).

2. **Context network.** The normalized output of the feature encoder is used as input in the Transformer context network to produce contextualized representations $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_T\}$:

$$g : \mathcal{Z} \mapsto \mathcal{C} \tag{5.2}$$

3. **Quantization module.** To enable self-supervised learning, the quantization module creates a finite set of speech representations $\mathbf{q}_t$ from the feature encoder outputs $\mathbf{z}$ using product quantization:

$$h : \mathcal{Z} \mapsto \mathcal{Q} \tag{5.3}$$

   Product quantization is used to generate targets for the self-supervised objective.

4. **Pre-training.** The model is pre-trained by masking parts of time steps from the feature encoder, and the objective is to identify the target quantized speech representation from a set of distractors.

5. **Masking.** Before using the feature encoder outputs as inputs in the context network, a part of the outputs are masked by a common trained feature vector. The inputs for the quantization module are not masked.

6. **Objective.** The objective during pre-training is to learn speech representations by optimizing a contrastive task $\mathcal{L}_m$ in addition with a codebook diversity loss $\mathcal{L}_d$ to equally use the codebook entries:

   $$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d \tag{5.4}$$

   The contrastive loss is defined as follows:

   $$\mathcal{L}_m = -\log \frac{exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} exp(sim(\mathbf{c}_t, \mathbf{q})/\kappa)} \tag{5.5}$$

   where $sim(\mathbf{c}, \mathbf{q})$ represents the cosine similarity between the context and the quantized speech representations. $\mathbf{c}_t$ is the output from the context network, $\mathbf{q}_t$ is the target quantized speech representation and $\mathbf{q}$ are the quantized candidate representations.

   The diversity loss $\mathcal{L}_d$ maximizes for each codebook $\bar{p}_g$ the entropy of the mean softmax distribution over its entries:

   $$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) \tag{5.6}$$

   where G and V represent the codebooks and codebook entries.

Then in the second step, the pre-trained model is fine-tuned for tasks such as speech recognition. A fully connected layer is introduced over the context network. This layer classifies the trained quantized speech representations into $\mathcal{C}$ classes which are specific to a language's character vocabulary. The Connectionist Temporal Classification (CTC) loss is used to optimize the models.

wav2vec 2.0 XLSR-53

The wav2vec 2.0 XLSR model was proposed by [4] and is based on the wav2vec 2.0 framework that pre-trains cross-lingual speech representations by pooling speech signals from multiple languages. wav2vec 2.0 XLSR-53 is a model checkpoint based on this methodology pre-trained on 53 languages from LibriSpeech, MLS, and BABEL. It was released by [4] to catalyze low-resource speech recognition research.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

6 Methodology

After introducing the background notions, we define our methodology to address our research questions. The different methodologies we apply make the distinction between the deliverables of this thesis and the outcomes from the previous work in [3]. In our previous work, we trained a monolingual model in one experiment and used transfer learning in another to enable speech recognition for Luxembourgish. These models were not fine-tuned on thoroughly normalized speech labels nor rescored by language modelling. In this study, we implement a more advanced pre-processing pipeline to normalize Luxembourgish speech labels. With the normalized transcriptions, we use force alignment from the aeneas library to segment transcribed audio files into utterances. Furthermore, we apply multilingual wav2vec 2.0 XLSR models that pre-train cross-lingual speech representations to improve our previous results. Finally, we will use an n-gram LM to rescore the decoding of our ASR models.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

Data Collection

For this thesis, we use labelled radio broadcast audio from the *Radio Telé Lëtzebuerg*[1] (RTL.lu) domain. Access to this data was given by the digital director of the media company to support the development of speech technologies for Luxembourgish. We reuse the audio-transcription pairs scraped during the study in [3]. This data consists of the *Apropos*, *Commentaire* and *Carte Blanche* radio shows. These radio shows invite guest speakers of the day to discuss the latest topics. These shows can be found under the *Opinion* - (Meenung) category on RTL.lu[2]. In our previous work, we created a scraping bot that gathers webpages from each radio emission category on RTL.lu that contained an audio file with its text transcription. Due to the GDPR [33], proper due diligence concerning data collection must be conducted. Therefore, only audio-transcription data pairs were collected without Personal Identifiable Information (PII), such as the

[1] https://www.rtl.lu/

[2] https://www.rtl.lu/meenung

[33], *2018 reform of EU data protection rules*, 2018.

speaker's name and gender. The raw labelled audio collected from the radio emissions amounts to 61 hours of labelled data. Each emission duration varies between 2 and 3 minutes. As unlabelled data, we collected in our previous work 842 hours of unlabelled speech from the same domain, although not strictly from the same radio emissions but could also originate from other sources such as interviews or television news broadcasts.

Text Normalization

In past approaches, Finite State Transducers (FSTs) were used to build grammars that handle text normalization. The trend for text normalization in recent literature is neural approaches instead of relying on linguistic knowledge to define hand-written language-specific grammars represented by FSTs [10–12]. Using neural approaches to model text normalization as a sequence-to-sequence problem works well for high-resource languages. However, this is not feasible for the Luxembourgish language since no Luxembourgish input-label pairs exist that represent the verbalizations of NSWs. Therefore, we rely on our Luxembourgish linguistic knowledge to define language-specific grammar in our text processing procedure.

[10] Mansfield et al., "Neural Text Normalization with Subword Units", 2019.

[11] Zhang et al., "Neural models of text normalization for speech applications", 2019.

[12] Yolchuyeva, Németh, and Gyires-Tóth, "Text normalization with convolutional neural networks", 2018.

Before creating language-specific grammar for our text processing pipeline, we have to analyze the text transcriptions. The transcription that comes with the audio from the radio emissions is useable for building ASR systems. However, in some cases, they are not very clean. After analyzing a sample of transcriptions of the audio files, we observe that most issues are related to the non-verbalizing of NSWs. The main issues are the verbalizations of numbers and standard abbreviations. Other issues are related to handling special characters or variations of the apostrophe.

In order to handle the expansion of types of numbers such as years, floating, ordinal, and cardinal numbers, we implemented the support for the Luxembourgish language in the number to words conversion library *num2words*[3]. We based our implementation on the German number system and introduced specific rules from Luxembourgish, such as the n-rule. The n-rule defines that the ending letter n of a word is not dropped if the next word begins with a consonant $c \in \{d, h, n, t, z\}$ or with a vowel $v \in \{i, u, e, o, a\}$.

In our text processing pipeline, we defined grammars as FSTs that map matched strings to desired strings. We implemented our FST grammars using regular expressions in Python to match string patterns and replaced them with the correct values. Our pre-processing pipeline is based on heuristics and may not represent a general and optimal solution to normalize the transcriptions. The pipeline starts

[3] https://github.com/letzspeak/num2words

by removing arbitrary URLs and verbalizing the dot in common Luxembourgish domain names.

As a second step in the pipeline, we standardize number expressions to prepare the text data in a consistent format for efficient number-to-word conversion. For example, some transcriptions format large numbers with commas that are placed at every third decimal place, while other transcription texts apply the inverse format that is used in many non-English speaking countries where we use periods to separate three decimal places and use commas as a separator between a number and its fractional part. We decided to remove all the number formatting to enable the automatic number expansion with the num2words library. In this step, we also have to handle numbers with postfixes $p \in \{'ten', 'sten', 'te'\}$. These postfixes mean that a number ending in them should be an ordinal number where the postfix needs to be substituted with a dot as defined in the Luxembourgish number system. As the last operation in this step, we introduced a space character between a number and a unit that follows it.

After this step, we have a standardized number formatting that can be processed by other grammar that uses num2words for conversion. Before expanding the different number types in the text data, we take care of the expansion of standard abbreviations. We defined a dictionary of common Luxembourgish abbreviations as a lookup table. This dictionary is used to substitute each occurrence of its abbreviation entry with the expanded form in the text.

After handling the abbreviation expansion, we treat the different number types in a fourth step, where we expand the different types of numbers in the following order: $years < ordinal\ numbers < floating\ numbers < cardinal\ numbers$. This way, it is easier to express the regular expressions for the subsequent substitution. For example, ordinal numbers and floating numbers are formatted with a dot in Luxembourgish and expanding first the ordinal numbers ensures that we do not match floating numbers as ordinal numbers.

Finally, in the last step, we verbalize special characters $c \in \{\%, +, \&\}$. Then, we remove non-printable characters or characters that are not valid in Luxembourgish. The last operation in this step is to normalize whitespace characters. We strip sequences of whitespace characters and substitute them with a single one.

Audio Segmentation of labelled and unlabelled data

After normalizing the transcription texts, we end up with a collection of Luxembourgish audio-transcription pairs that are not noisy as training data. As a next step, we present our approach to segment the transcribed audio into smaller utterances that can be used to fine-tune the acoustic model. In recent studies, [7, 9], alignment algorithms such as the Smith-Waterman common subsequences identification algorithm are often used to segment long audio files according to text transcriptions to create training data for speech technologies. However, this approach does not scale well for under-resourced languages. In many under-resourced languages, no speech recognition models are available to generate audio transcription and align it with the target text.

Our approach uses force alignment to match speaker utterances from an audio file to their candidate labels extracted from the audio transcription. Force alignment has the same task as the alignment approach using common subsequences identification. Contrary to the alignment algorithm from [7, 9] that uses reference and candidate text pairs, force alignment, used from the *aeneas*[4] library, generates segmented utterances by chunking the transcription text into text fragments first. Then each text fragment is synthesized with a Text-to-Speech (TTS) engine. To align the synthesized text fragment with the reference signal, we extract MFCCs from both signals and apply the Sakoe-Chiba Band DTW algorithm. This procedure generates sync maps in JSON format, and they represent timestamps of the segment boundaries. However, force alignment in aeneas does not estimate the confidence level within a region of similarity as in the approaches of [7, 9]. Without a confidence metric evaluating the correctness of a segment label, we need language experts to validate the labelled utterances. Therefore, for this study, we implement the idea of the human in the loop similar to the data processing pipeline from [6].

We design a tool where a user can validate aligned audio-transcription pairs and correct the transcription if necessary. Visualizations of this validation tool is shown in Figure 6.1 and 6.2. This validation tool is implemented using the React[5] library to create a web application. We designed the interface as user-friendly as possible to improve the user experience while validating the transcribed alignments. Figure 6.1 illustrates the index page of the web application. It summarizes in the top UI component the main statistics about the current validation, such as the total number of radio emissions in the corpus, total duration of the audio collection and validated hours of audio. Additionally, it lists each radio emission on this page. When a user clicks on an audio item, the application redirects the user to the detail view of the audio item.

A detailed view of a segmented audio item is pictured in Figure 6.2. The detail view has many UI elements. First, it visualizes the waveform of the current audio to the user and highlights the currently selected segment area within the waveform. Underneath the

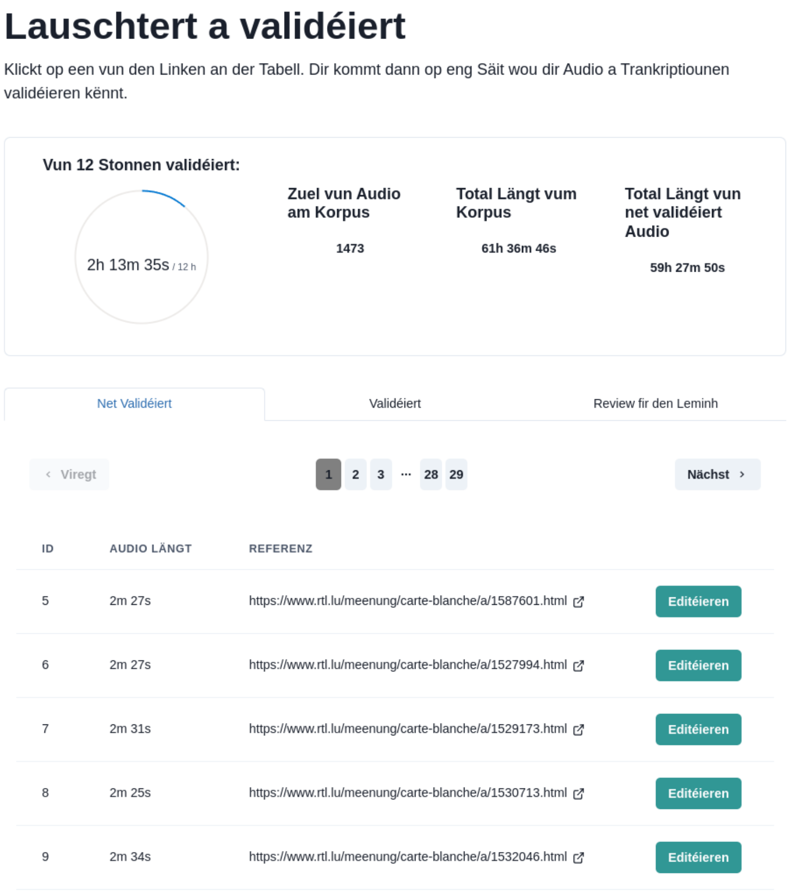
[7] Pratap et al., “MLS: A large-scale multilingual dataset for speech research”, 2020.

[9] Panayotov et al., “Librispeech: an ASR corpus based on public domain audio books”, 2015.

[4] https://www.readbeyond.it/aeneas/

[6] Ardila et al., “Common Voice: A Massively-Multilingual Speech Corpus”, 2020.

[5] https://reactjs.org/



Lauschtert a validéiert

Klickt op een vun den Linken an der Tabell. Dir kommt dann op eng Säit wou dir Audio a Trankriptiounen validéieren kënnt.

Vun 12 Stonnen validéiert:

2h 13m 35s / 12 h

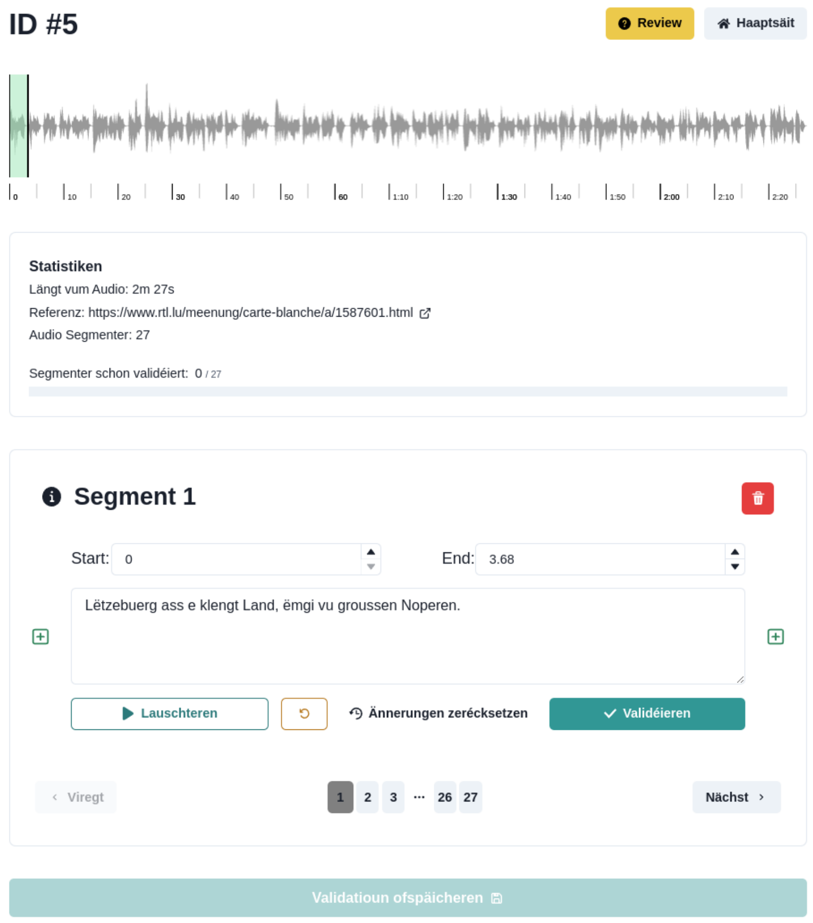
Zuel vun Audio am Korpus	Total Längt vum Korpus	Total Längt vun net validéiert Audio
1473	61h 36m 46s	59h 27m 50s

ID	AUDIO LÄNGT	REFERENZ	
5	2m 27s	https://www.rtl.lu/meenung/carte-blanche/a/1587601.html	Editéieren
6	2m 27s	https://www.rtl.lu/meenung/carte-blanche/a/1527994.html	Editéieren
7	2m 31s	https://www.rtl.lu/meenung/carte-blanche/a/1529173.html	Editéieren
8	2m 25s	https://www.rtl.lu/meenung/carte-blanche/a/1530713.html	Editéieren
9	2m 34s	https://www.rtl.lu/meenung/carte-blanche/a/1532046.html	Editéieren

Figure 6.1: Index page of the audio segments validation tool.

waveform component, we list different information about the audio to the user. The information box displays the audio file duration and the source from where it was collected. Besides this information, we show the user the total number of segments. Additionally, a progress bar tells the user how many segments are already validated. The third visual component is the editor area, where the user can correct the current transcription, start and end timestamps. In this editor area, we provide the user with the functionalities to play back or validate the current segment. Furthermore, the user can create and delete segments.

The final validation tool was hosted and distributed to the team at the Zenter fir d'Lëtzebuerger Spooch to have language experts validate the transcribed audio segments. The colleagues decided to standardize the orthography of the speech labels only to use the spelling of the main variation of a word. Reaching our goal of validating 12 hours of audio segments, we continued to increase our validation until 14 hours of labelled speech. Following our efforts to validate and



ID #5

Statistiken
Längt vum Audio: 2m 27s
Referenz: https://www.rtl.lu/meenung/carte-blanche/a/1587601.html
Audio Segmenter: 27

Segmenter schon validéiert: 0 / 27

Segment 1

Start: 0 End: 3.68

Lëtzebuerg ass e klengt Land, ëmgi vu groussen Noperen.

Figure 6.2: Detail page of the audio segments validation tool.

correct the labelled audio segments, we used the *pydub*[6] library to divide the large audio files using the validated sync maps containing the segment timestamps. The segmentation process yielded 6874 correctly transcribed utterances with an average duration of 7 seconds. The duration distribution of the audio segments is shown in Figure 6.3.

To divide the 842 hours of unlabelled audio files into 15 to 20 seconds long segments, we used the same procedure from [7]. In this procedure, we split the long audio files on silence intervals between 15 and 20 seconds. If no silence period was found during this interval, then we split the audio segment on the 20th second mark and set the new segment starting point at this split. We repeat this procedure until we reach the end of the audio file.

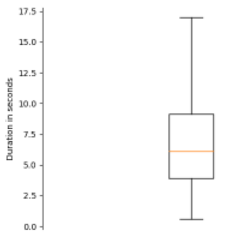


Figure 6.3: Box plot of duration distribution of 14 hours of labelled samples.

[6] https://pypi.org/project/pydub/

[7] Pratap et al., "MLS: A large-scale multilingual dataset for speech research", 2020.

Speech Corpus Creation

After successfully segmenting the collection of large audio files into smaller utterances, we need to split the utterances into different sets for training, validating and testing our speech recognition model. The training set is used to fine-tune the acoustic model for the downstream speech recognition task, while the validation set measures how well the model generalizes on unseen data and optimizes the model during the training step. The testing set evaluates the model's actual performance after training. To create the speech corpus, we shuffle the utterances first and allocate 95% to the training set. The rest 5% is evenly split between the validation and testing sets.

Furthermore, for our experiments, we created four training data configurations to study the effect of the amount of labelled data on the performance of wav2vec 2.0. These configurations represent training sets of different sizes that are 4, 8, 11 and 14 hours, respectively. The validation and testing sets are kept the same for each configuration. For the scope of this study, we are not ensuring speaker generalization. Thus, we shuffle the speech samples before splitting them into the different sets without ensuring that each speaker's utterances appear only in one data split. After splitting the samples into three sets, we persist the splits as TSV[7] files. To fine-tune a pre-trained wav2vec 2.0 model in Hugging Face, we format our training data as a Hugging Face *dataset* object[8]. We use the dataset creation script[9] and specify it to read the created TSV files in order to load the labelled speech samples for each data split. This script acts as a data loading script when called during fine-tuning or evaluating the model.

Replication of Baseline Results

Before fine-tuning our multilingual wav2vec 2.0 XLSR-53 models, we need to replicate the results from our previous study to set a baseline to compare our new models. In [3], we trained the first Luxembourgish wav2vec 2.0 models. One model was pre-trained from scratch only on the 842 hours of unlabelled speech. In another experiment, we trained the same unlabelled speech on a wav2vec 2.0 model pre-trained on 960 hours of English speech from the LibriSpeech corpus. Both models were introduced with a linear layer to fine-tune them on 4 hours of Luxembourgish speech from broadcast news using the CTC loss. The two models achieved a validation WER of 25.1% and 23.5%, respectively. In the previous work, the models were trained using Meta AI's *fairseq* library [40]. For the replication, we convert the fairseq wav2vec 2.0 models to the Hugging Face model implementation for the ease of model loading and inference enabled by their

[7] Tab Separated Values

[8] https://huggingface.co/docs/datasets/index

[9] https://github.com/huggingface/datasets/blob/master/templates/new_dataset_script.py

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

[40] Ott et al., "fairseq: A Fast, Extensible Toolkit for Sequence Modeling", 2019.

abstraction interfaces. Then with our validation and testing datasets combined with the Hugging Face evaluation Python script[10], we evaluate the converted checkpoints from our previous work on speech utterances from the same domain. The evaluation of the models shows a similar validation WER of 23.95% and 23.39%, respectively, although with a lower WER as obtained in the previous study. Table 6.1 shows the validation and testing evaluation results. This replication represents our baseline to benchmark our multilingual models with LM rescoring against it.

Model	Unlabelled data	Language Model	WER		CER	
			dev	test	dev	test
Baseline:						
4h labelled						
Base wav2vec 2.0	LB-842	None	23.95	23.09	7.97	7.63
Base wav2vec 2.0	LS-960 + LB-842	None	23.39	22.57	8.15	7.60

Table 6.1: CER and WER replication of our previous wav2vec 2.0 models on the Luxembourgish dev/test sets. The models were pre-trained using the audio from the Luxembourgish dataset (LB-842) as unlabelled data. The model was pre-trained on audio from LibriSpeech (LS-960) in the transfer learning setup.

Multilingual Speech recognition with wav2vec 2.0 XLSR

In the previous study, we pre-trained Luxembourgish speech representation from scratch. Additionally, we explored transfer learning from English latent representations in a second experiment. Self-supervised learning of speech representation is still widely used in recent high and low-resource speech recognition research. Thus, we continue to experiment with this paradigm. We investigate the self-supervised learning of cross-lingual speech representations to verify the claim of [4, 14] that learning of cross-lingual speech representations significantly outperform monolingual wav2vec 2.0 models pre-trained only on a single language. We evaluate a wav2vec 2.0 XLSR-53 fine-tuned with unlabelled Luxembourgish speech to prove the claim. For this evaluation, we have access to this model checkpoint trained on the HPC facilities at the University of Luxembourg [41] while I still had access during my undergraduate studies. This model is based on the wav2vec 2.0 XLSR model that was pre-trained on 53 languages from LibriSpeech, MLS, and BABEL. It was released by [4] to catalyze low-resource speech recognition research. The wav2vec 2.0 XLSR-53 was fine-tuned on 842 hours of unlabelled speech from the RTL.lu domain. Pre-training Luxembourgish speech representations in addition to the 53 languages took three days on four Nvidia V100 GPUs with 32 GB VRAM each. The pre-training visualization is shown in Figure 6.4. We convert this checkpoint to load the model weights in the Hugging Face wav2vec 2.0 implementation.

[4] Conneau et al., “Unsupervised Cross-lingual Representation Learning for Speech Recognition”, 2020.

[14] Babu et al., “XLS-R: Self-supervised cross-lingual speech representation learning at scale”, 2021.

[41] Varrette et al., “Management of an Academic HPC Cluster: The UL Experience”, 2014.

[10] https://github.com/huggingface/transformers/blob/main/examples/research_projects/robust-speech-event/eval.py

To fine-tune the multilingual wav2vec 2.0 model, we use the existing fine-tuning Python script[11] from the Hugging Face Robust Speech Challenge.

[11] `https://github.com/huggingface/transformers/blob/main/examples/research_projects/robust-speech-event/run_speech_`

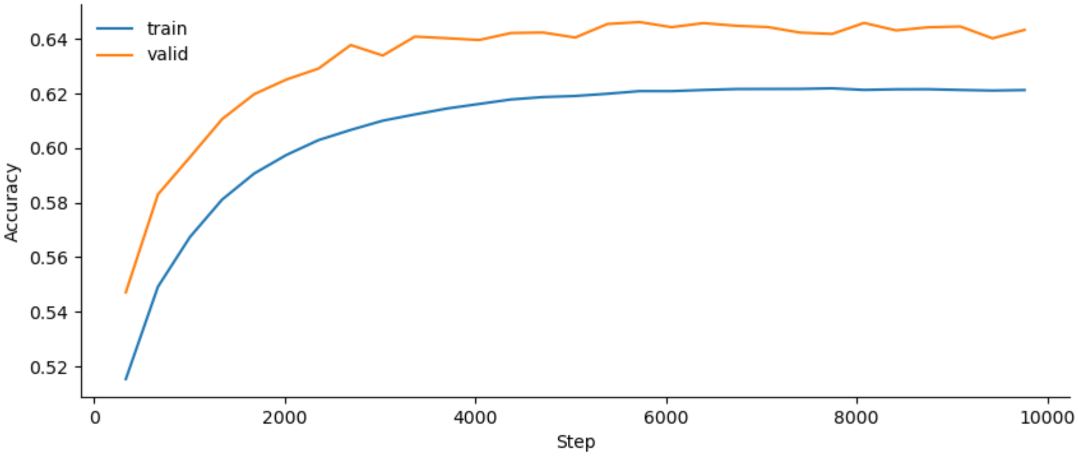


Figure 6.4: Accuracy visualization while pre-training Luxembourgish Speech Representations on top of wav2vec 2.0 XLSR-53 checkpoint.

We fine-tuned the pre-trained acoustic model on different labelled data setups of 4, 8, 11, and 14 hours. All our fine-tuning experiments were executed on a mobile Nvidia 1070 GPU with 8 GB of VRAM, and the duration lasted between 12 hours to 1.5 days. The training and validation loss of the 14 hours fine-tuning experiment is visualized in Figure 6.5. In Figure 6.6, we visualize the validation WER during our fine-tuning experiment with 14 hours of labelled speech.

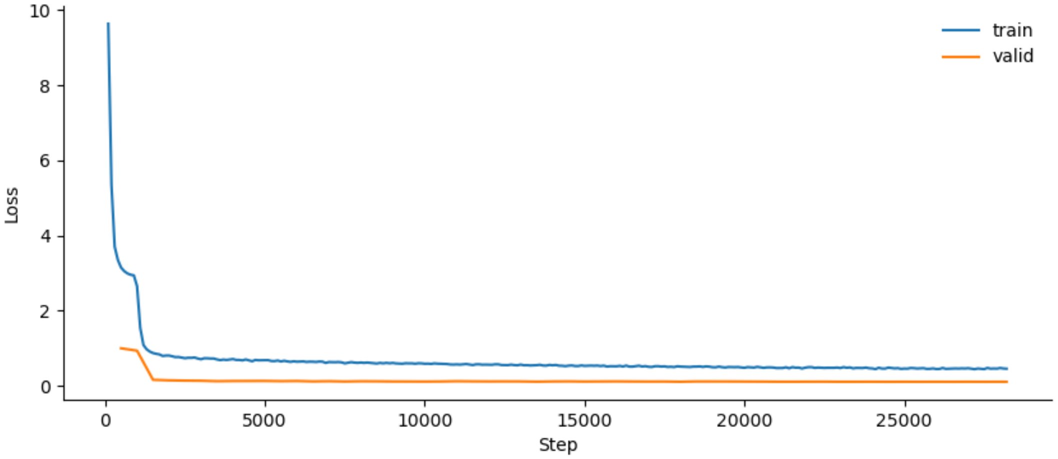


Figure 6.5: Training and validation loss visualization while fine-tuning multilingual wav2vec 2.0 XLSR-53 checkpoint with 14 hours of labelled speech.

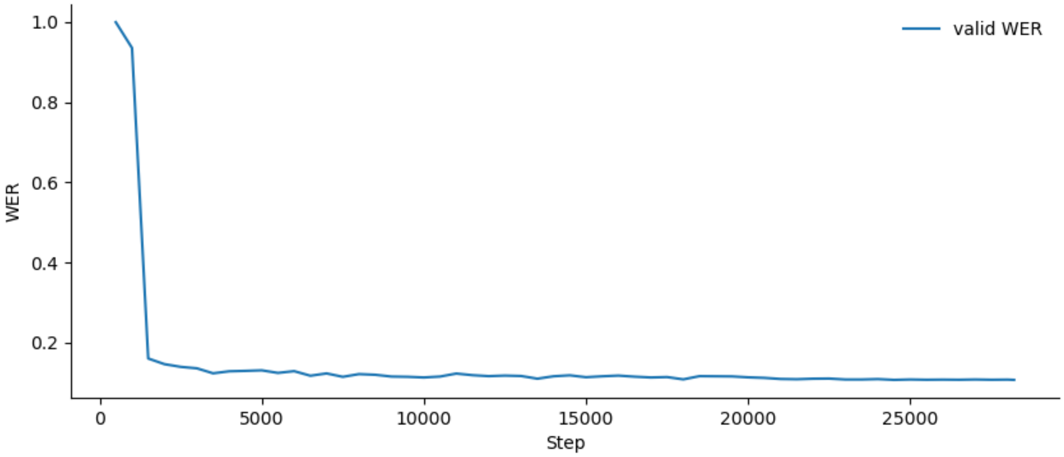


Figure 6.6: Validation WER visualization when fine-tuning on 14 hours of labelled speech.

Language Model Rescoring using n-grams

We rely on n-gram LMs to reestimate our speech decodings. Even though language modelling based on neural network architectures are outperforming their n-gram counterpart, neural-based LMs are still very computationally expensive [27, 28, 30, 36]. Additionally, we decided for n-gram language modelling since it is well integrated into the Hugging Face wav2vec 2.0 implementation. The methodology for augmenting wav2vec 2.0 models with LM rescoring is based on their blog article from [36]. First, we need to obtain large text resources since n-gram LMs estimate word sequence likelihoods from a reference text corpus. For this task, we obtained access to two text corpora. ZLS provided access to the parliamentary debate transcriptions from 2020 to 2022, representing 4 million tokens. The second corpus was given to us by RTL.lu and represents 16 million tokens from their news articles domain. In total, the text corpus corresponds to 20 million tokens. Our experiments explore which corpus domain performs better when rescoring the model outputs. To build the LM, we use the raw text in most cases and apply only the removal of special characters from the text corpus and standardize variations of the apostrophe. We used the *KenLM*[12] library to create a 5-gram model based on our text corpus with the installed binaries. A 5-gram is an n-gram that estimates the likelihood of 5-word sequences. Then we followed the rest of the specific code snippets from [36] to incorporate the LM into the decoder of our fine-tuned wav2vec 2.0 model. These code snippets handle the missing start of sentence tokens in the 5-gram models and combine the KenLM 5-gram model into the

[27] Xu et al., "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition", 2018.

[28] Kumar et al., "Lattice rescoring strategies for long short term memory language models in speech recognition", 2017.

[30] Arisoy et al., "Bidirectional recurrent neural network language models for automatic speech recognition", 2015.

[36] Platen, *Boosting Wav2Vec2 with n-grams in HF Transformers*, 2022.

[12] https://github.com/kpu/kenlm

Hugging Face wav2vec 2.0 implementation.

7 Results and Discussion

We first evaluate our first experiment where we pre-trained a wav2vec 2.0 XLSR-53 on 842 hours of unlabelled Luxembourgish speech and fine-tuned the model with 4 hours of labelled speech. This experiment is equivalent to our baseline models, using the same amount of labelled data to fine-tune the models.

Table 7.1: Results of our Luxembourgish wav2vec 2.0 fine-tuning experiments. CER and WER on the Luxembourgish dev/test sets when training on the labelled data setups of 4h, 8h, 11h and 14h. The models were pre-trained using the audio from the Luxembourgish dataset (LB-842) as unlabelled data. In the transfer learning setup, the model was pre-trained on audio from LibriSpeech (LS-960). The best WERs and CERs on dev and test sets are highlighted in bold.

Model	Unlabelled	Language	WER		CER	
	data	Model	dev	test	dev	test
Baseline:						
4h labelled						
Base wav2vec 2.0	LB-842	None	23.95	23.09	7.97	7.63
		5-gram	20.10	18.67	7.41	6.76
Base wav2vec 2.0	LS-960 + LB-842	None	23.39	22.57	8.15	7.60
		5-gram	18.40	17.75	7.15	6.74
This work:						
4h labelled						
XLSR-53 wav2vec 2.0	LB-842	None	19.44	18.77	7.16	6.43
		5-gram	16.11	15.10	6.63	5.79
8h labelled						
XLSR-53 wav2vec 2.0	LB-842	None	13.86	12.87	3.11	2.91
		5-gram	10.94	10.32	2.48	2.39
11h labelled						
XLSR-53 wav2vec 2.0	LB-842	None	12.68	11.53	2.76	2.55
		5-gram	9.98	10.09	2.27	2.22
14h labelled						
XLSR-53 wav2vec 2.0	LB-842	None	11.68	10.71	2.64	2.31
		5-gram	**9.50**	**9.30**	**2.17**	**2.08**

The evaluation of this model on the test set shows that learning cross-lingual representations are essential for low-resourced languages such as Luxembourgish. Learning cross-lingual representations and rescoring the output transcriptions with language modelling while using the same amount of labelled speech achieves a WER of 15.1% and improves the previous best result for Luxembourgish speech recognition from [3] relatively by 33.1% and absolutely by 7.5%. Increasing the amount of labelled speech yields a significant performance gain. Our best multilingual wav2vec 2.0 model fine-tuned on 14 hours of speech reaches a 9.3% WER. The detailed results of our fine-tuning experiments with different training data sizes are reported in Table 7.1.

[3] Nguyen, *Self-Supervised Learning of Speech Representations*, 2021.

We visualize the findings of our fine-tuning experiments in Figure 7.1 which showcases better the data from Table 7.1. The larger the transcribed dataset increases, the more the WER trend of the speech decoding decreases. We observe the same trend when decoding with language modelling. Overall the 5-gram LM decreases the WER between one and two per cent.

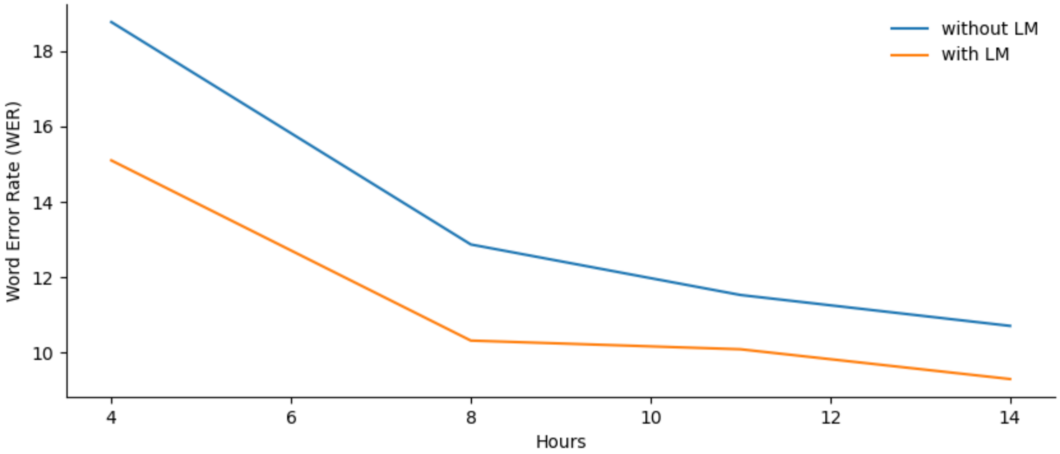


Figure 7.1: Word Error Rate respective to labelled dataset size. With and without Language Modelling (LM).

Additionally, in our language modelling experiments, we study the impact of the text corpus size on recognition performance. In language modelling, the perplexity metric measures how well an LM predicts a sequence of words. A low perplexity metric indicates a good model. The perplexity of an LM is related to the sparseness of a reference text corpus used to train the model. To reduce the perplexity, we generally increase the corpus size or apply word decomposition algorithms to reduce the out-of-vocabulary rate. Therefore, we investigate the impact of two 5-gram LMs on speech recognition decoding. The first LM is trained on 4 million tokens from the parliament transcription corpus. The second model is trained on the combination of the parliament and RTL.lu text resources, totalling 20 million tokens. In Figure 7.2, we plot the WERs for each LM decoder used in our different-sized acoustic models.

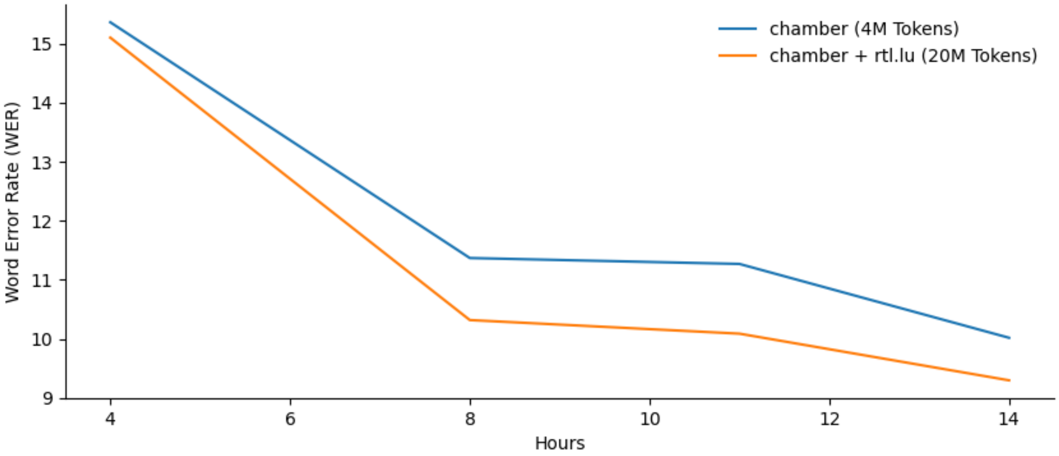


Figure 7.2: Word Error Rate respective to language model size.

In the first configuration, where the acoustic model was fine-tuned on only 4 hours of labelled speech, the increase in text data did not improve the WER considerably. However, when the size of the labelled dataset increases, the larger text corpus significantly decreases the WER for the acoustic models fine-tuned with 8 and 11 hours, respectively. This observation could imply that using more text data for language modelling is less effective in low resource than in high resource acoustic models. Additionally, we can interpret that using text resources from the same domain as the audio training data benefits the improvement of the speech decoding performance.

To analyze the errors emitted by the speech decoder, we inspect the model inference on the test dataset and compare it to the ground truth labels. We can summarize the common decoding errors into the following categories:

1. Words composed of multiple words are decoded as separate words. E.g. *schlësselroll* is decoded by the model as *schlëssel roll*

2. The speaker pronounces loan words from contact languages how they would sound in Luxembourgish. E.g. *caféepicerie* which is a loan word from French, is decoded by the model as *kaffi episserie*.

3. The speaker pronounces a correct variation of a word that is not used in the ground truth. E.g. *interessi* is pronounced instead of *interesse*.

4. The n-rule is often misused while speaking. The letter *n* at the end of words is not dropped when speaking even though it is dropped in the written form.

5. On some occasions, the article *d'* is missing in the output transcription.

6. The speaker pronounces silent vowels and consonants.

7. The model decodes the wrong vowel. E.g. *d'schwemm* instead of *d'schwämm*.

We addressed our research questions with our methodology to verify our hypotheses. Our results show that pre-training cross-lingual speech representations combined with LM rescoring improve the Luxembourgish speech recognition performance.

The first research question explores if pre-training cross-lingual representations could improve wav2vec 2.0 models that have been pre-trained on Luxembourgish solely. The wav2vec 2.0 XLSR model [4] is a multilingual speech recognition model that pre-trains on cross-lingual speech representations. Experiments have shown that cross-lingual pre-training significantly outperforms monolingual pre-training. Therefore, we hypothesized, following [4] that pre-training cross-lingual representations will improve monolingual wav2vec 2.0 models that have been trained solely on Luxembourgish. Our experimental results show that Luxembourgish acoustic models benefit from cross-lingual pre-training and improve our baseline models' recognition performance. This validates our hypothesis and the claim from [4] that pre-training cross-lingual representations improve the ASR performance for low-resourced languages.

[4] Conneau et al., "Unsupervised Cross-lingual Representation Learning for Speech Recognition", 2020.

The second research question investigates if LM rescoring could improve the baseline Luxembourgish speech recognition models that use the Greedy algorithm for decoding. Experiments were conducted with LM rescoring in the original wav2vec 2.0 [2] paper and significantly improved the model's performance. Following [24], we hypothesized that using LMs for rescoring will further improve the Luxembourgish wav2vec 2.0 models. Our LM rescoring experiments show that Luxembourgish speech recognition models benefit from language modelling, and by increasing the text corpus, the speech recognition performance improves significantly. Our experiments validate our hypothesis and the claim from [24] that incorporating an LM to rescore the speech decoder improves the recognition accuracy by emitting the best recognition hypothesis.

[2] Baevski et al., *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020.

[24] Besacier et al., "Automatic speech recognition for under-resourced languages: A survey", 2014.

8 Conclusion

We presented our improvements over our previous work on Luxembourgish wav2vec 2.0 models trained in a monolingual and transfer learning setting. We investigated the self-supervised multilingual learning of Luxembourgish speech representations to be used for the downstream speech recognition task. An extensive literature review was performed to frame our research questions and hypotheses within the field. We introduced a reproducible methodology to address our research questions and verify our hypotheses. Our methodology describes our procedure to pre-process our collected speech labels. Then we segmented the transcribed audio into utterances by applying force alignment combined with validation from language experts. Finally, we fine-tuned a wav2vec 2.0 XLSR-53 checkpoint pre-trained on 842 hours of unlabelled Luxembourgish speech.

Our experiments validate our hypotheses that learning cross-lingual representations and LM rescoring are essential for low-resourced languages such as Luxembourgish. Learning cross-lingual representations and rescoring the output transcriptions with language modelling while using only 4 hours of labelled speech achieves a word error rate of 15.1% and improves the previous best result for Luxembourgish speech recognition relatively by 33.1% and absolutely by 7.5%. Increasing the amount of labelled speech to 14 hours yields a significant performance gain resulting in a 9.3% word error rate.

We expect performance improvements by collecting more Luxembourgish text resources for the language modelling. In addition, to collecting more text data, we can now use our ASR model to augment the transcribed speech corpus by applying the Smith-Waterman alignment algorithm. This approach inexpensively increases the Luxembourgish speech corpus since there are publicly available speech-transcription pairs, and we do not need to rely on language experts to supervise audio segmentation and correct speech labels.

With wav2vec 2.0 enabling speech recognition for Luxembourgish with little labelled data and our collaboration with the Zenter fir d'Lëtzebuerger Sprooch and RTL.lu, we expect to implement this solution in production to make Luxembourgish speech recognition accessible to end-users and improve the digital inclusion in Luxembourg. We hope to catalyze research in natural language processing for Luxembourgish by generating more text resources from broadcast news with our speech recognition system.

Bibliography

[1] Martine Adda-Decker et al. "Developments of "Lëtzebuergesch" Resources for Automatic Speech Processing and Linguistic Studies". In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. 2008.

[2] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL].

[3] Le Minh Nguyen. *Self-Supervised Learning of Speech Representations*. 2021.

[4] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning for Speech Recognition". In: *CoRR* abs/2006.13979 (2020). arXiv: 2006.13979. URL: https://arxiv.org/abs/2006.13979.

[5] Mark JF Gales et al. "Speech recognition and keyword spotting for low-resource languages: Babel project research at cued". In: *Fourth International workshop on spoken language technologies for under-resourced languages (SLTU-2014)*. International Speech Communication Association (ISCA). 2014, pp. 16–23.

[6] R. Ardila et al. "Common Voice: A Massively-Multilingual Speech Corpus". In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*. 2020, pp. 4211–4215.

[7] Vineel Pratap et al. "MLS: A large-scale multilingual dataset for speech research". In: *arXiv preprint arXiv:2012.03411* (2020).

[8] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. "Low-resource languages: A review of past work and future challenges". In: *arXiv preprint arXiv:2006.07264* (2020).

[9] Vassil Panayotov et al. "Librispeech: an ASR corpus based on public domain audio books". In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 5206–5210.

[10] Courtney Mansfield et al. "Neural Text Normalization with Subword Units". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 190–196. DOI: 10.18653/v1/N19-2024. URL: https://aclanthology.org/N19-2024.

[11] Hao Zhang et al. "Neural models of text normalization for speech applications". In: *Computational Linguistics* 45.2 (2019), pp. 293–337.

[12] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. "Text normalization with convolutional neural networks". In: *International Journal of Speech Technology* 21.3 (2018), pp. 589–600.

[13] Hanan Aldarmaki et al. "Unsupervised automatic speech recognition: A review". In: *Speech Communication* (2022).

[14] Arun Babu et al. "XLS-R: Self-supervised cross-lingual speech representation learning at scale". In: *arXiv preprint arXiv:2111.09296* (2021).

[15] Tessfu Geteye Fantaye, Junqing Yu, and Tulu Tilahun Hailu. "Investigation of automatic speech recognition systems via the multilingual deep neural network modeling methods for a very low-resource language, Chaha". In: *Journal of Signal and Information Processing* 11.1 (2020), pp. 1–21.

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

[17] Hirunika Karunathilaka et al. "Low-resource sinhala speech recognition using deep learning". In: *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE. 2020, pp. 196–201.

[18] Cheng Yi et al. "Applying wav2vec2.0 to speech recognition in various low-resource languages". In: *arXiv preprint arXiv:2012.12121* (2020).

[19] Brij Mohan Lal Srivastava et al. "Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages." In: *SLTU*. 2018, pp. 11–14.

[20] Karel Veselý et al. "Lightly Supervised vs. Semi-supervised Training of Acoustic Model on Luxembourgish for Low-resource Automatic Speech Recognition". In: *Proc. Interspeech 2018*. 2018, pp. 2883–2887. DOI: 10.21437/Interspeech.2018-2361. URL: http://dx.doi.org/10.21437/Interspeech.2018-2361.

[21] Febe De Wet et al. "Speech recognition for under-resourced languages: Data sharing in hidden Markov model systems". In: *South African Journal of Science* 113.1-2 (2017), pp. 1–9.

[22] Jia Cui et al. "Multilingual representations for low resource speech recognition and keyword search". In: *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*. IEEE. 2015, pp. 259–266.

[23] Martine Adda-Decker, Lori Lamel, and Gilles Adda. "Speech alignment and recognition experiments for Luxembourgish". In: *Spoken Language Technologies for Under-Resourced Languages*. 2014.

[24] Laurent Besacier et al. "Automatic speech recognition for under-resourced languages: A survey". In: *Speech communication* 56 (2014), pp. 85–100.

[25] Samuel Thomas et al. "Deep neural network features and semi-supervised training for low resource speech recognition". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 6704–6708.

[26] Viet-Bac Le and Laurent Besacier. "Automatic speech recognition for under-resourced languages: application to Vietnamese language". In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.8 (2009), pp. 1471–1482.

[27] Hainan Xu et al. "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition". In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2018, pp. 5929–5933.

[28] Shankar Kumar et al. "Lattice rescoring strategies for long short term memory language models in speech recognition". In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2017, pp. 165–172.

[29] William Chan et al. "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2016, pp. 4960–4964.

[30] Ebru Arisoy et al. “Bidirectional recurrent neural network language models for automatic speech recognition”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 5421–5425.

[31] Roland Kuhn and Renato De Mori. “A cache-based natural language model for speech recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 12.6 (1990), pp. 570–583.

[32] Temple F Smith and Michael S Waterman. “Identification of common molecular subsequences”. In: *Journal of molecular biology* 147.1 (1981), pp. 195–197.

[33] *2018 reform of EU data protection rules*. European Commission. 2018. URL: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes%5C_en.pdf.

[34] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[35] Thomas Pellegrini and Lori Lamel. “Are audio or textual training data more important for ASR in less-represented languages?” In: *Spoken Languages Technologies for Under-Resourced Languages*. 2008.

[36] Patrick von Platen. *Boosting Wav2Vec2 with n-grams in HF Transformers*. https://huggingface.co/blog/wav2vec2-with-ngram, accessed: 22/06/22. 2022.

[37] Aston Zhang et al. *Dive into Deep Learning*. https://d2l.ai. 2020.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[39] Lilit Yolyan. *Review on Self-Supervised Contrastive Learning*. https://towardsdatascience.com/review-on-self-supervised-contrastive-learning-93171f695140, accessed: 10/06/21. 2021.

[40] Myle Ott et al. “fairseq: A Fast, Extensible Toolkit for Sequence Modeling”. In: *Proceedings of NAACL-HLT 2019: Demonstrations*. 2019.

[41] S. Varrette et al. “Management of an Academic HPC Cluster: The UL Experience”. In: *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*. Bologna, Italy: IEEE, July 2014, pp. 959–967. URL: https://hpc.uni.lu.