UNIVERSITETI I EVROPËS JUGLINDORE
УНИВЕРЗИТЕТ НА ЈУГОИСТОЧНА ЕВРОПА
SOUTH EAST EUROPEAN UNIVERSITY

DOCTORAL DISSERTATION TOPIC

A FRAMEWORK FOR ALBANIAN SPEECH RECOGNITION USING DEEP LEARNING TECHNIQUES

CANDIDATE:

Msc. Amarildo Rista

MENTOR:

Prof.Dr. Arbana Kadriu

Tetovo, September 2022

Declaration,

I state that this doctorate thesis comes from the results of my research at the Department of Contemporary Sciences and Technologies (CST), at the Southeast European University (SEEU).

References for other studies as well as reports of any other scholar are properly referenced. This doctoral dissertation, complete or partial, was not presented or submitted for any other degree.

Name: Amarildo Rista

Signed: 

Date: 18 September 2022

ABSTRACT

Speech recognition is a discipline of natural language processing (NLP) that generates methodologies and technologies to recognize and translate speech into text by a machine. It reduces the gap in human-machine communication. In developed countries and for high-resource languages many studies have been done in this direction. But in this field, it is difficult to standardize a model applicable to all languages in the world, since each language has its own grammatical, morphological, syntactic, and semantic features, as well as the features of the spoken language such as dialects, socio-linguistic and ethnocultural characteristics, or even physical specifics of the personal vocal system, are different.

Today, the development of deep learning technology is considered one of the biggest achievements of machine learning techniques. Deep Learning is a multi-layered neural network designed to imitate the human brain. It has achieved state-of-the-art performance in the application of automatic speech recognition (ASR) systems. This is derived from the predictive power of deep neural networks. The success of deep learning in the domain of speech recognition started with the presentation of end-to-end (E2E) models. The End-to-end ASR systems integrate the language model (LM), the acoustic model (AM), and the pronunciation model (PM) into a neural network. This optimizes components jointly by improving the overall performance of the model.

Recently, Transformers have made an extraordinary turn in speech recognition and beyond. Transformers is a powerful deep learning model which works through sequence-to-sequence learning. They use an attention mechanism that provides context around items in the input sequence. So rather than start to run the first word of the sentence, the Transformers attempt to identify the context that brings meaning to each word of the sequence. And it is this attention mechanism that gives Transformers a huge leg up over algorithms like RNN that must run in sequence.

In this thesis, first, we design a corpus for the Albanian language, suitable for training and evaluating ASR systems. It consists of 100 hours of audio recordings along with their transcripts. And second, we investigate the Albanian language's speech recognition (SR) as a low-resource scenario of automatic speech recognition (ASR), by exploring various methods and architectures based on deep learning. Design of architectures for Speech Recognition of the Albanian language will be focused on the Deep

Speech models that are recurrent neural network (RNN) based architectures with different approaches, and Transformers-based, which will be built in the Pytorch tool.

The main goal of this research (thesis) is to design and implement a framework for Albanian Speech Recognition, to be able to recognize Albanian speech generated from each person with state-of-the-art performance.

The evaluation of the model will be done through the metrics of word error rate (WER) and character error rate (CER). Results showed that our proposed models achieve great performance. The best architecture achieves very satisfactory WER and CER results, where the WER goes to 5% and CER to 1% on the training set and 18% WER and 11% CER on the testing set.

This study introduces a noble contribution to the field of natural language processing, with, a focus on the Albanian language, which is expected to accelerate research within this domain.

Keywords: *Speech Recognition, CASR corpus, Albanian Language, Deep learning, Automatic Speech Recognition, end-to-end, Transformers*.

ABSTRAKT

Njohja e ligjeratës është një disiplinë e përpunimit të gjuhës natyrore (NLP), që gjeneron metodologji dhe teknologji për të njohur dhe përkthyer të folurin në tekst nëpërmjet kompjuterit. Kjo disiplinë redukton hendekun në komunikimin njeri-makinë. Në vendet e zhvilluara dhe për gjuhët me burime të larta janë bërë shumë studime në këtë drejtim. Por në këtë fushë është e vështirë të standardizohet një model i zbatueshëm për të gjitha gjuhët në botë, pasi çdo gjuhë ka veçoritë e veta gramatikore, morfologjike, sintaksore dhe semantike, si edhe veçoritë e gjuhës së folur si dialektet, karakteristikat sociale dhe kulturore, apo edhe specifikat fizike të sistemit vokal janë të ndryshme.

Sot, zhvillimi i teknologjisë deep learning konsiderohet si një nga arritjet më të mëdha në machine learning. Deep learning është një rrjet neural me shumë shtresa i krijuar për të imituar trurin e njeriut. Ai ka arritur performancën më të mirë në aplikimin e sistemeve të njohjes automatike të të folurit (ASR). Kjo rrjedh nga fuqia parashikuese e rrjeteve të thella neurale. Suksesi i deep learning në fushën e njohjes së të folurit filloi me prezantimin e modeleve end-to-end (E2E). Sistemet ASR end-to-end integrojnë modelin gjuhësor (LM), modelin akustik (AM) dhe modelin e shqiptimit (PM) në një rrjet neural. Kjo optimizon komponentët së bashku duke përmirësuar performancën e përgjithshme të modelit.

Kohët e fundit, Transformers kanë bërë një kthesë të jashtëzakonshme në njohjen e të folurit dhe më gjerë. Transformers është një model i bazuar në deep learning, i cili funksionon përmes mësimit sekuencë për sekuencë. Ai përdor një mekanizëm “vëmendjeje” që jep kuptimin e të gjithë sekuencës hyrse. Pra, në vend që të fillojë të procesojë fjalën e parë të fjalisë, Transformers përpiqet të identifikojë kontekstin që sjell kuptim për cdo fjalë të sekuencës. Ky mekanizëm i jep Transformers një hap të madh mbi algoritmet si RNN që duhet të funksionojnë në sekuencë.

Në këtë tezë, së pari, ne krijojmë një korpus për gjuhën shqipe, të përshtatshme për trajnimin dhe vlerësimin e sistemeve ASR. Ai përbëhet nga 100 orë regjistrime audio së bashku me transkriptet e tyre. Dhe së dyti, ne hulumtojmë njohjen e të folurit të gjuhës shqipe (SR) si një skenar me burime të ulëta të njohjes automatike të të folurit (ASR), duke eksploruar metoda dhe arkitektura të ndryshme të bazuara në deep learning. Dizajni i arkitekturave për njohjen e të folurit të gjuhës shqipe do të fokusohet në

modelet deep speech që janë arkitektura të bazuara në rrjetin neural të përsëritur (RNN) me qasje të ndryshme dhe në modelet e bazuara në Transformers, të cilat do të ndërtohen në mjetin Pytorch.

Qëllimi kryesor i këtij studimi (teze) është të hartojë dhe zbatojë një kornizë për njohjen e të folurit shqip, që të jetë në gjendje të njohë të folurin shqip të gjeneruar nga çdo person me performancë të lartë.

Vlerësimi i modelit do të bëhet përmes shkallës së gabimit të fjalës (WER) dhe shkallës së gabimit të karakterit (CER). Rezultatet treguan se modelet tona të propozuara arrijnë performancë të shkëlqyer. Arkitektura më e mirë arrin rezultate shumë të kënaqshme të WER dhe CER, ku WER shkon në 5% dhe CER në 1% në grupin e trajnimit dhe WER shkon ne 18% dhe CER ne 11% në grupin e testimit.

Ky studim paraqet një kontribut fisnik në fushën e përpunimit të gjuhës natyrore, me fokus në gjuhën shqipe, e cila pritet të përshpejtojë kërkimet në këtë fushë.

Fjalë Kyçe: *Njohja e të folurit, korpusi CASR, Gjuha Shqipe, Deep Learning, Njohja automatike e të folurit, end-to-end, Transformers.*

АПСТРАКТ

Препознавањето говор е дисциплина на обработка на природниот јазик (NLP) која генерира методологии и технологии за препознавање и преведување на говор во текст преку машина. Го намалува јазот во комуникација човек-машина. Во развиените земји и за јазиците со високи ресурси многу студии се направени во оваа насока. Меѓуто, тешко е да се стандардизира модел применлив за сите јазици во светот, бидејќи секој јазик има свои граматички, морфолошки, синтактички и семантички карактеристики, како и карактеристиките на говорниот јазик како што се дијалекти, социо-лингвистички и етнокултурни карактеристики, па дури и физичките специфики на личниот вокален систем, се различни.

Денес, развојот на технологијата за длабоко учење се смета за едно од најголемите достигнувања на техниките за машинско учење. Deep Learning е повеќеслојна невронска мрежа дизајнирана да го имитира човечкиот мозок. Има постигнато најсовремени перформанси во примената на автоматскиот систем за препознавање говор. Ова е изведено од моќта на предвидување на длабоките невронски мрежи. Успехот на длабокото учење во доменот на препознавање говор започна со презентација на end-to-end (E2E) моделите. Системите ASR од крај до крај го интегрираат јазичниот модел (LM), акустичниот модел (AM), и моделот на изговор (PM) во невронска мрежа. Ова ги оптимизира компонентите заедно со подобрувањето на вкупните перформанси на моделот.

Неодамна, Трансформаторите (Transformers) направија извонреден пресврт во препознавањето говор и пошироко. Трансформаторите се моќен модел за длабоко учење кој работи преку учење од редослед до секвенца. Тие користат механизам за внимание што обезбедува контекст околу ставките во влезната низа. Така попрво отколку да започне да го изведува првиот збор од реченицата, трансформаторот се обидува да го идентификува контекстот што му дава значење на секој збор од низата. И токму овој механизам за внимание им дава на Трансформаторите огромен чекор над алгоритмите како RNN кои мора да работат во низа. Во оваа теза, прво, дизајниравме корпус за албанскиот јазик, погоден за обука и оценување ASR системи. Овој корпус се состои од 100 часа аудио снимки заедно со нивните транскрипти. И второ, ние го истражуваме препознавањето говор на албанскиот јазик (CP) како сценарио со ниски ресурси за автоматско

препознавање говор (ASR), со истражување на различни методи и архитектури засновани на длабоко учење.

Дизајнот на архитектури за препознавање на говор на албанскиот јазик ќе биде фокусиран на длабокото говорни модели кои се архитектури базирани на рекурентна невронска мрежа (RNN) со различни пристапи, и базирани на трансформатор, кои ќе бидат изградени со алатката Pytorch.

Главната цел на ова истражување (теза) е да се дизајнира и имплементира рамка за која може да го препознае албанскиот говор генериран од кој било човек, со најсовремени перформанси. Оценувањето на моделот ќе се врши преку метрика на стапка на грешка на зборовите (WER) и стапка на грешка на карактерите (CER). Резултатите покажаа дека нашите предложени модели постигнуваат одлични перформанси. Најдобрата архитектура постигнува многу задоволителни резултати на WER и CER, каде што WER оди до 5%, а CER до 1% на сетот за обука и 18% WER и 11% CER на комплетот за тестирање.

Оваа студија воведува значаен придонес во областа на обработката на природните јазици, со фокус на албанскиот јазик, што се очекува да го забрза истражувањето во овој домен.

Клучни зборови: *препознавање на говор, корпусот CASR, албански јазик, длабоко учење, автоматско препознавање говор, end-to-end, трансформатор*.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my thesis supervisor, Prof. Dr Arbana Kadriu. She guided and encouraged me throughout this journey, to be professional even when the road was full of obstacles. She was always available and responsible, whenever I needed her treasured instructions. Without her persistent help, this project would not have been realized.

I would like to pay my special regards to my colleagues and friends who have stood by me during this journey.

I wish to express my deepest gratitude to my parents and sisters for their unconditional support during all the years of my studies and during this study cycle. Without their support and love, this achievement would not have been possible.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

ASR	Automatic Speech Recognition
NLP	Natural Language Processing
RNN	Recurrent Neural Network
ResCNN	Residual Convolutional Neural Networks
BiRNN	Bidirectional Recurrent Neural Networks
WER	Word Error Rate
CER	Character Error Rate
AM	Acoustic Model
LM	Language Model
CASR	Corpus of Albanian Language
LSTM	Long Short-term Memory
BLSTM	Bidirectional Long Short-term Memory
GRU	Gated Recurrent Unit
CTC	Connectionist Temporal Classification
CNN	Convolutional Neural Network
DBN	Deep Belief Networks
MFCC	Mel-Frequency Cepstral Coefficient

fMLLR Feature Space Maximum Likelihood Linear Regression

HMM Markov Hidden Model

GMM Gaussian Mixture Model

SGMM Subspace Gaussian Mixture Model

MLP Multilayer Perceptron

SOM Self-Organizing Map

DNN Deep Neural Network

E2E End-to-End

LDA Linear Discriminant Analysis

MLLT Maximum Likelihood Linear Transform

FFN Feed-forward Network

TDNN Time Delay Neural Network

MHA Multi-head Attention

LAS Listen, Attend and Spell

ARSG Attention-based Recurrent Sequence Generator

LVCSR Large Vocabulary Continuous Speech Recognition

RNN-T Recurrent Neural Network-Transducer

GCNN Graph Convolutional Neural Network

GLU Gated Linear Unit

GeLU Gaussian Error Linear Units

VTLP Vocal Tract Length Perturbation

WRCNN Wide Residual Convolutional Layer

1. INTRODUCTION

Automatic speech recognition (ASR) has grown exponentially in recent years. It is considered a useful tool that significantly overcomes the gap in human-computer and computer-human interaction. An ASR system recognizes the speech and translates it to text through a machine. It is composed of four main modules: 1) signal processing and feature extraction; 2) acoustic model (AM); 3) language model (LM); 4) and hypothesis search (Yu and Deng, 2016).

Today, the ASR systems are classified into three basic categories: 1) traditional hybrid systems (Dahl et al., 2011), (Graves et al., 2013), (Sun et al., 2020); 2) end-to-end systems (Graves and Jaitly, 2014), (Amodei et al., 2016), (Chan et al., 2016); and 3) Transformers-based ASR systems (Vaswani et al., 2017), (Dong et al., 2028), (Gulati et al., 2020).

The traditional hybrid systems have dominated during the last decade. These systems are based on classical Hidden Markov Models (HMM) (Juang and Rabiner, 1991) and Gaussian Mixture Models (GMM) (Reynolds, 2009) models, which require forced aligned data. Although these systems are still widely used today, they have a few drawbacks. Low accuracy is the biggest disadvantage. In addition, each component of the model should be trained independently, and a significant amount of human labour is needed.

An end-to-end ASR system integrates the language model (LM), acoustic model (AM), and pronunciation model (PM) into a neural network. The joint optimization of all these components optimizes the model's overall performance. These models are based on Recurrent Neural Network (RNN) (Mandic and Chambers 2001) and their advanced versions such as CNN (O'Shea and Nash 2015), BiRNN (Kamath et al. 2019), ResCNN (Vydana and Vuppala 2017), etc. These systems convert a sequence of input features into a sequence of words without force-aligned data. In addition, these systems are easy to train, require less human labour, and have great accuracy.

Transformers marked an extraordinary turning point in ASR systems. Transformers are a powerful deep learning model which works through sequence-to-sequence learning, where the Transformers take a sequence of tokens as input, and predict the next word in the output sequence. What makes Transformers a little bit different is that they do not necessarily process data in order. Transformers use a mechanism called attention, which provides context around items in the input sequence. So rather than start to run the first word of the sentence, the Transformers attempt to identify the context that brings meaning to each word of the sequence. And, it is this attention mechanism that gives Transformers a huge leg up over traditional systems as well as end-to-end systems. Transformers run multiple sequences in parallel and this accelerates training time.

One of the biggest challenges for an ASR system is its adaptation to the speaker's attributes such as voice, accent, dialect, and speed of speech. In addition, there are environmental conditions (clean, noisy) as well as audio recording equipment. An ASR system should be robust to these variables to produce the correct transcripts.

The development of ASR systems is based on corpus-driven techniques. Well-annotated speech corpus is the desirable quality of spoken language resources for the development and evaluation of ASR systems. So, a quality corpus should be checked manually with human intervention. To create a large quality corpus for research and development purposes, we need to give a great effort referring to speech data collection, annotation, validation, organization, and documentation.

This thesis addresses two main issues which are undoubtedly interconnected to each other. The first issue consists of the design of a Corpus for Albanian Speech Recognition (CASR), aimed at training and evaluating SR models for the Albanian language. The corpus consists of 100 hours of audio recordings jointly with their transcripts. It includes a rich vocabulary that covers the topics such as biography, social and political sciences, psychology, religion, economics and business, history, philosophy, and sociology, to be as heterogeneous as possible. During the design of the corpus, we considered the phonetic, semantic, morphology, and syntax features of the Albanian language. In addition, we have considered the speaker's features such as gender, age, voice, and dialect, which are very important in a corpus. The second issue consists of the investigation of the speech recognition (SR) of Albanian by exploring various methods and architectures based on deep learning. Design of architectures for SR of the Albanian

language mainly is focused on the end-to-end Deep Speech models as well as Transformers-based approaches.

To evaluate the accuracy of the model, Word Error Rate (WER) and Character Error Rate (CER) metrics will be used.

1.1 Importance of the Thesis

Over the years, developed countries and high-resource languages have developed relatively fast toward natural language processing (NLP), specifically in speech recognition. In this field, it is difficult to standardize a model applicable to all languages in the world, since each language has its own grammatical, morphological, syntactic, and semantic features, as well as the features of the spoken language such as dialects, socio-linguistic and ethnocultural characteristics, or even physical specifics of the personal vocal system are different.

This thesis addresses some of these issues, by focusing on the Natural Language Processing of the Albanian Spoken Language.

This thesis presents the design of a Corpus for Albanian Speech Recognition (CASR), aimed at training and evaluating various ASR models and will culminate in the development of various architectures for Speech Recognition of the Albanian language.

This study introduces a noble contribution to the domain of Natural Language Processing, which is expected to accelerate research within the speech community for the Albanian language.

1.2 Research Aims and Objectives

This research aims to design and implement a framework for Albanian Speech Recognition. The study introduces:

- Design and creation of a corpus for the Albanian language which will be used to train and evaluate different ASR systems.
- Design and implementation of various architectures for Speech Recognition of the Albanian language focused on Deep Learning techniques.

The main goal of this research (thesis) is to design and implement a framework for Albanian Speech Recognition, to be able to recognize Albanian speech generated from each person with state-of-the-art performance.

The objectives of the thesis are as follows:

I. Review of the most cited and popular ASR architectures during the last decade.
II. Building a corpus in the Albanian language suitable to train and evaluate Albanian ASR systems.
III. Development of the specific tools to support ASR architectures of the Albanian language.
IV. Training and evaluation of various ASR architectures to achieve state-of-the-art WER results.

1.3 Research Questions

The research questions have been mapped by the research aims and objectives of this study. This thesis address three research questions as follow:

I. Are deep learning techniques suitable for speech recognition of the Albanian language?
II. How do corpus specifications (size, voice, and dialect) affect the accuracy of the model?
III. Which deep learning-based architecture for Albanian speech recognition achieves state-of-the-art performance?

- To answer the first research question, we will analyze the results of the experiments for Albanian and English languages as well as results extracted by literature review.
- To answer the second research question, first, we will analyze several corpora of different sizes and second, we will analyze two corpora, where one is built with the standard Albanian language by a single speaker and the other is built with both dialects of the Albanian language and by several speakers of both genders and different age groups.

- To answer the third research question, we will analyze the experimental results for all proposed models in this study.

1.4 Research Hypotheses

I. The deep learning techniques are suitable for Albanian speech recognition.
II. The corpus specifications (size, voice, and dialect) affect the accuracy of the model.

III. The end-to-end deep learning model for the Albanian language achieves state-of-the-art WER (Word Error Rate) results.

If “H1” is true, the implicit conclusion would be that deep learning techniques need to consider a long design of speech recognition systems for the Albanian language.

If “H2” is true, the implicit conclusion would be that corpus specifications need to consider a long design of corpus.

If “H3” is true, the implicit conclusion would be that end-to-end models could be used as an opportunity to achieve state-of-the-art WER in SR systems for the Albanian language.

1.5 Research Methodology

Our research will answer the hypothesis and research questions addressed. It includes three main steps: Design of speech recognition architectures for the Albanian language; Design and creation of a corpus for the Albanian language; Training and evaluation of the proposed models. Figure 1, provides a graphical presentation of the methodology workflow.

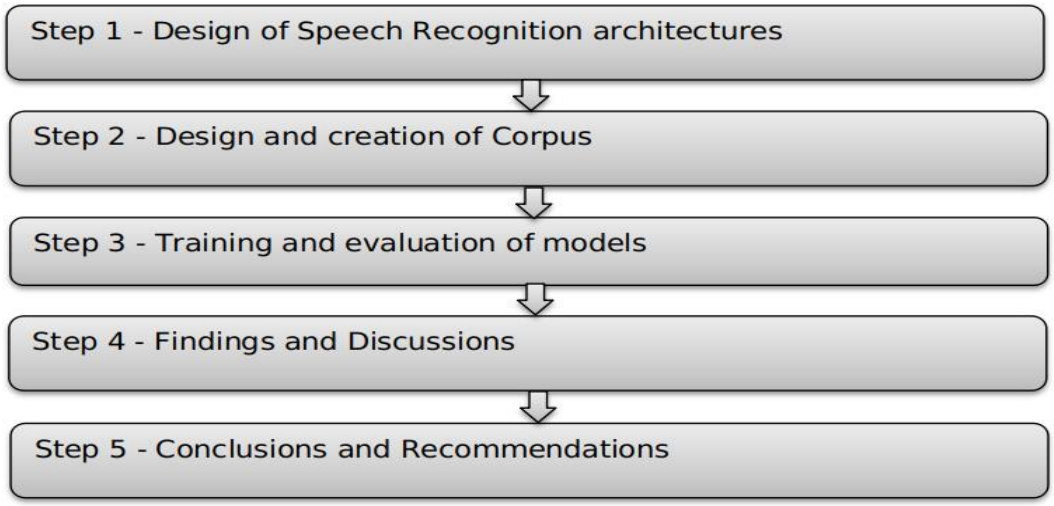

Figure 1. Methodology workflow.

Step 1 provides the building of architectures for Speech Recognition of the Albanian language, which will be focused on end-to-end approaches and Transformers-based models. The end-to-end architectures

will be focused mainly on RNN-based models and their advanced approaches such as BiRNN, ResCNN, and GRU. While Transformers-based architectures will be focused on wav2vec approaches.

Step 2 provides the design and creation of a corpus for Albanian Speech Recognition (CASR), aimed at training and evaluating ASR models. The corpus will be focused on various sources of data, comprising 100 hours of transcribed audio data in the Albanian standard language as well as in both dialects.

Step 3 provides training and evaluation of the proposed models based on research questions and hypotheses addressed in this study. The evaluation will be done in both the training set and testing set to measure the accuracy of the models for both known and unknown data.

Step 4 provides the research findings and discussions, while step 5 provides the research conclusion and recommendations of the study.

1.6 Structure of the Thesis

Chapter 1 outlines the background and introduction of the research study; problem definition; aims and objectives; research questions and hypothesis; methodology and significance of the study.

Chapter 2 of the thesis presents the literature review focused fundamentally from an academic point of view. First, it outlines the background and introduction to ASR systems. It continues with the identification of appropriate research papers. We present a survey of Hybrid ASR techniques; End-to-End ASR techniques; ASR Systems for Low Resource Languages; and Transformers-based ASR systems. The last part of Chapter 2, continues with an overview of the Albanian Language and, finally, we present the results extracted by literature.

Chapter 3 explains the methodology which has been used to carry out the research. It also includes the model's design, corpus development, tools, and assessment criteria.

Chapter 4 presents all experiments and the results of the study. Initially, it presents the evaluation of the proposed end-to-end models through the training set and the testing set. Then it continues with the evaluation of the effect of CASR corpus specifications on the accuracy of the model as well as the evaluation of CASR in comparison to LibriSpeech corpus.

In the last part, the evaluation of the Transformers-based model through the testing set has been presented.

Chapter 5 represents the discussions and findings of the research. Initially, it presents findings derived from the literature review, called secondary findings. Then it continues with findings derived from our experiments called primary findings.

Chapter 6 is the last section of this study. It presents the conclusions and an overview of all our work in this thesis along with our plans for future activities.

1.7 Conclusion

In this chapter, we outlined the background and introduction to the research study. Then we stated the importance of this thesis.

In Section 1.2, the research aims and objectives were addressed. The main goal of this thesis is to design a framework for Albanian ASR, to be able to recognize Albanian speech produced by any speaker.

In Section 1.3 the research questions have been mapped by the research aims and objectives of the study.

In Section 1.4, we addressed three hypotheses to answer the research questions targeted and to help us, what we should look for in the experimental phase.

Finally, in Section 1.5, we have presented the methodology workflow of our study.

2. LITERATURE REVIEW

Researchers have shown different perspectives related to Speech Recognition. In this section of the thesis, appropriate research papers have been dealt with. After the introduction, the methodology for the analysis of SR research and studies is presented, followed by various Speech Recognition Systems and an overview of the Albanian language.

2.1 Introduction

Automatic speech recognition (ASR) improves human-computer interactions. Today, this technology has changed our lives and it became one of the primary means for humans to interact with much other equipment. There are many applications in which speech recognition plays an important role such as health care (Latif et al., 2020), education (Dalim et al., 2020), industry (Vajpai et al., 2016) and many internets of things (Abdulkareemand et al., 2021) machine learning applications (Helmke et al., 2020).

An ASR system consists of four main modules: 1) signal processing and feature extraction; 2) acoustic model (AM); 3) language model (LM); 4) and hypothesis search (Yu and Deng, 2016), as is shown in Figure 2.

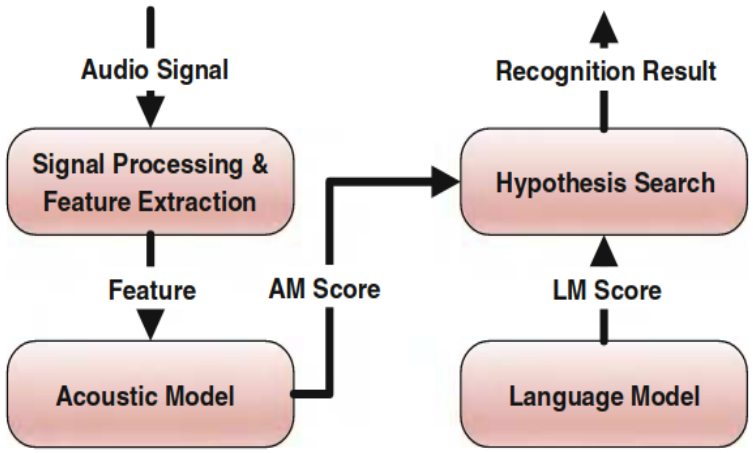

Figure 2. The architecture of ASR systems (Yu and Deng, 2016).

The first module is fed by audio waves in the time domain, extracts the linguistic content, removes background noise and irrelevant information and transforms it to the frequency domain.

The AM is fed by the features extracted from the first module and outputs a result for phonemes or other linguistic units with different sequential lengths.

The language model (LM) models the word sequences by determining the word probability through an algorithm that establishes rules for context by analyzing the corpus to predict new sentences.

The last module is the hypothesis search, which folds the acoustic model, language model, and lexicon to generate the transcripts.

During the last decade, ASR systems were mainly based on classical Machine Learning technologies such as Hidden Markov Model (HMM) (Eddy, 1996) and Gaussian Mixture Model (GMM) (Reynolds, 2009). The combination of these models with each other well as with the most advanced technologies, form what are today called Traditional HMM and GMM systems. In these systems, each component must be trained independently and require forced aligned data. Although these systems are widely used today, they have some disadvantages. The first is the low accuracy of the model, which requires a custom phonetic group to improve. Second, training each component of the model independently requires a lot of time and is labour intensive. And thirdly, the forced aligned data are difficult to obtain and access.

In 2014, Hannun et al published the paper "Deep Speech: Scaling up end-to-end speech recognition" which marked a turning point in the field of ASR systems. They proposed an end-to-end ASR system which applies deep learning techniques. This system directly maps a sequence of speech data into a sequence of words and does not require forced-aligned data. In addition, all components are trained jointly. The most popular end-to-end ASR systems are based on deep learning architectures such as CTC (Watanabe et al., 2017), LAS (Chan et al., 2016), RNN (Miao et al., 2015), CNN (Palaz et al., 2015) and various approaches of them. Some of the advantages that these systems have compared to hybrid systems are: reduction of training and decoding time, simplicity in implementation, require less human labour, and most importantly offering greater accuracy.

Nowadays, Transformers have changed radically the development of ASR systems. Transformers is a powerful deep learning model which is composed of an encoder and a decoder (Vaswani et al., 2017). It works through sequence-to-sequence learning where the Transformers take a sequence of tokens as input and predicts the next word in the output sequence. The innovation of the Transformers consists in the application of the attention mechanism which provides context around items in the input sequence. In this way instead of processing the first word of the sentence, the Transformers attempt to identify the context that brings meaning to each word of the sequence. In addition, this attention mechanism gives Transformers a huge leg up over algorithms like RNN that must run in sequence. Transformers run multiple sequences in parallel and this vastly speeds up training times.

2.2 Methodology

This chapter presents a comprehensive study, investigating various technologies applied in ASR systems. For this purpose, we have searched using a keyword-based strategy in the most popular databases such as Google scholar, Scopus, Elsevier, ACM, Crossref, etc., to identify the most cited and popular papers in the domain of Speech Recognition. The target keywords used are : "automatic speech recognition (ASR)", "hybrid ASR system", "end-to-end ASR system", "transformers ASR", "low-resource language", "corpus", "Albanian language", "deep learning". In total, we have screened about 300 papers, and we have selected to analyze over 150 papers.

The performed research study includes the following steps: The first step presents the most cited and popular hybrid ASR systems of the last decade. In the second step, the most popular end-to-end ASR systems are identified. The third step consists of the identification of low-resource ASR systems. The fourth step identifies Transformers-based ASR systems, and the last step presents an overview of the Albanian Language. Figure 3 shows the methodology workflow.

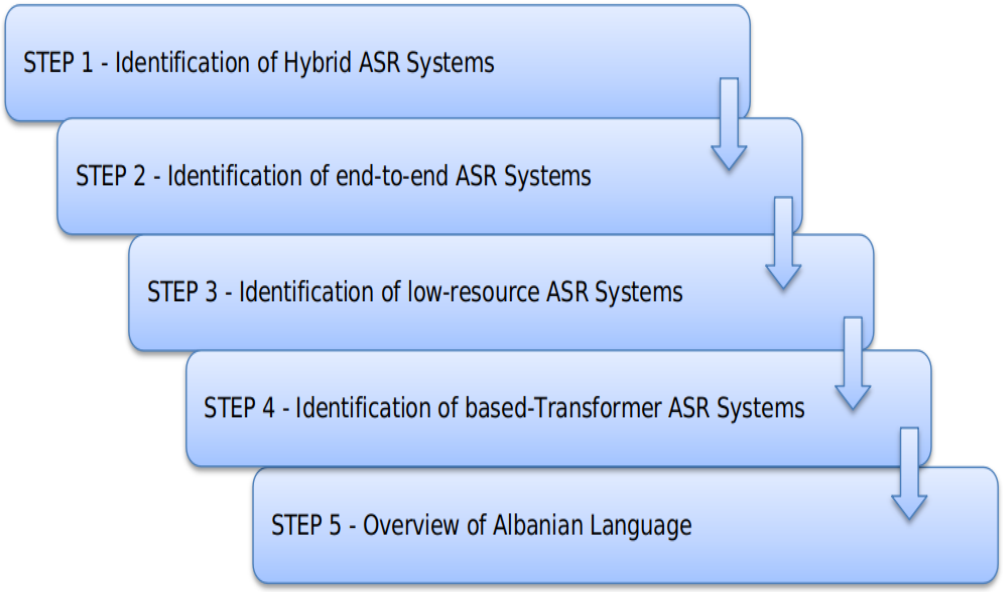

Figure 3. Methodology Work-flow.

2.3 Hybrid Speech Recognition Systems

Hybrid ASR systems are mainly based on classical HMM and GMM models. Despite their shortcomings, these systems combined with Neural Networks (NNs) and various deep learning approaches as well as Transformers, are widely used today.

In this session, we will analyze the most cited and popular hybrid ASR architectures during the last decade. For each selected architecture, we will make a detailed analysis by describing all the components with which it is built. In addition, we will make a description of the way of training and testing each architecture, highlighting at the end of each architecture its performance. To analyze the performance of each architecture, we have referred to the international standard word error rate (WER) parameter.

At the end of this session, we aim to define which hybrid architecture performs better, to serve us as a good reference to continue with our study towards the definition of a high-performance architecture for speech recognition of the Albanian language.

One of the most cited hybrid systems is presented by Dahl et al (2012). They propose a context-dependent (CD) model, which combines the power of learning of deep neural networks (DNNs) with sequential modelling of HMMs called CD-DNN-HMM. This system models the audio wave through HMM sequentially and evaluates the probability of predicted words through DNNs. Figure 4 presents the baseline architecture of this model.

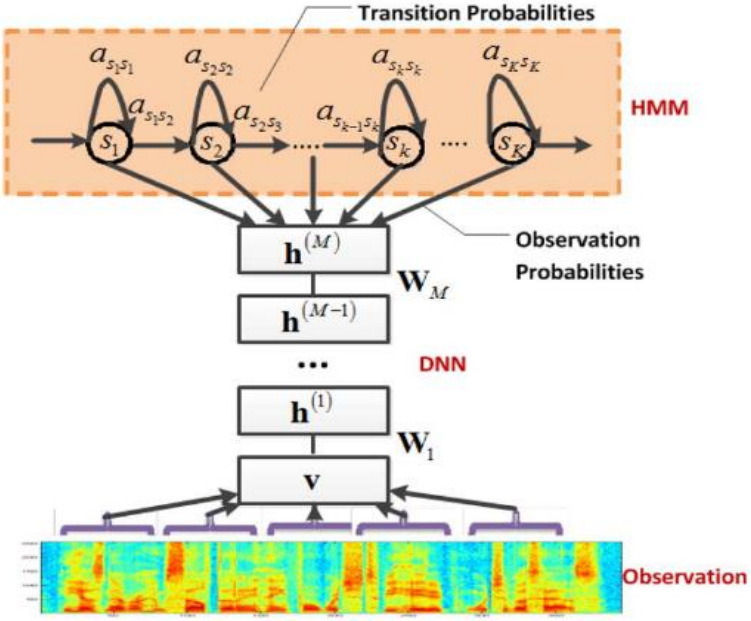

Figure 4. The architecture of the DNN-HMM hybrid system (Dahl et al, 2011).

The innovation of this approach is modelling senone is performed by evaluating their probability through deep neural networks by using only a single audio sequence. Training the model through the Viterbi algorithm is one of the strengths of this approach. Also, it provides quite an efficient decoding. This model achieves a WER of 29.7% on the test set, by overcoming the performance of existing architectures in this domain.

Jinyu Li et al (2012) have proposed an improved version of CD-DNN-HMM baseline architecture, using mixed bandwidth training data to improve prediction accuracy. In this way, they reduce the mixed-

bandwidth training problem into a missing feature problem by designing the filter bank (Hossan et al., 2010), as the input to the DNNs.

Three main components are added to this approach that is decisive in improving its accuracy.

I. Modelling senones directly regardless of their large number, which can be tens of thousands.
II. Using deep neural networks (DNNs) that have extraordinary predictive power instead of multi-layer perceptron (MLPs).
III. Using a long context window of frames as the input to the DNNs.

This approach improves the performance by outperforming the baseline CD-HMM-DNN architecture by 3-4 %, by achieving 27.47 % of WER.

Jaitly et al (2012) have proposed a Deep Belief Networks (DBN) pre-trained context-dependent ANN/HMM system. The ANN-HMM baseline architecture computes probabilities of speech data using Bayes rules to predict the HMM features. The novelty of this approach is the integration of a multi-layered DBN model with ANN-HMM. DBNs are composed of some layers of neural networks, which are known as Restricted Boltzmann Machines (Fischer and Igel, 2012). Every Boltzmann machine is restricted to a single visible layer and a hidden layer, which connect the previous layer with the ensuing layers and it ensures that nodes within a group do not interact with each other. Application of DBN increases the number of hidden layers increasing the performance of the system in terms of training time as well as accuracy, regardless of the size of the corpus. The training process of the DBN-ANN-HMM system goes through three phases:

I. In the first phase, after the basic HMM-GMM architecture is trained, the forced alignment rule is applied to connect the data sequence with the HMM feature targeted.
II. In the second phase, training of the DBN through MFCC or filter bank vectors is performed.
III. In the third phase, the features of the DBN create a DNN to predict the HMM features extracted by the speech data.

This hybrid system exceeds performance expectations by achieving a WER of 11.8%. But, there are some drawbacks such as hardware requirements, a huge corpus to achieve better performance, and it is quite complex looking for hundreds of machines to perform.

Another ASR hybrid system is presented by Kadyan and Kaur (2020). This approach models the acoustic model (AM) on two front-end GFCC and MFCC approaches. Figure 5 presents the block diagram of the GMM-HMM approach.

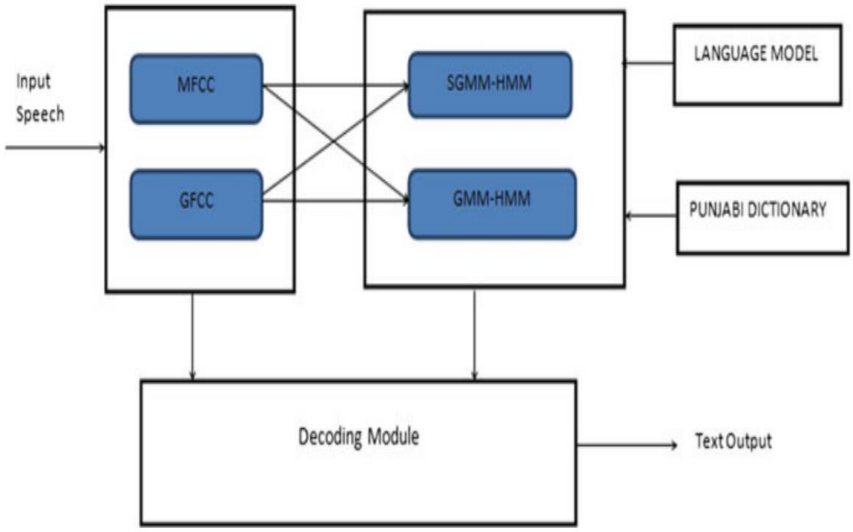

Figure 5. Block diagram of SGMM-HMM-based Punjabi ASR system (Kadyan and Kaur, 2020).

The workflow follows these steps:

I. Generation of GFCC and MFCC feature vectors extracted by input speech signal.

II. Computation of CVNM.

III. Training of the monophone model to produce graphs.

IV. Alignment of the monophone output model to produce its graph on the triphone model.

V. Production of data graphs by applying LDA and MLLT.

VI. Computation of training data based on SAT technique on LDA+MLLT alignment.

VII. Data analysis.

Referring to experimental results, the SGMM-HMM approach overcame the GMM-HMM with an improvement of 3 - 4%. Also, it overcomes the problem of sharing state parameter information throughout the training and testing of a system.

Sun and Chol (2020) proposed a hybrid system which combines the longer context information of recurrent neural networks (RNN) with the subspace Gaussian mixture model (SGMM) to model an acoustic model suitable for ASR. It analyzes the vector states of a word using longer context information of RNN to model each word from an SGMM. To model, the acoustic model has applied the fMLLR algorithm for adaptation of SGMLM as described by (Povey et al., 2011). The pipeline of this framework goes through three phases:

I. Training of the recurrent neural network (RNN) based language model through the corpus, which outputs context features of a word in continuous space. These features capture longer context information and feed subspace Gaussian mixture-based language models (SGMLM) training it.

II. Construction of acoustic models based on Gaussian subspace mixture models. So that the system is as robust as possible, the authors use decision trees in the Top-Down method and an agglomerative clustering approach in the Bottom-Up method as described by (Babich et al., 1996) for parameter tying. This is very important since many words in the corpus repeat themselves.

III. Adaptation of SGMLM with fILLER to improve system performance. This adaptation consists in solving two major problems faced by language models. First, solving the problem of inconsistency between training data and testing data by applying the fMLLR-based language model adaptation. And the second is the processing of unseen data of the training set in the language model using RNN.

Figure 6 describes the construction of the SGMLM block diagram, by presenting in detail phases 2 and 3 described above.

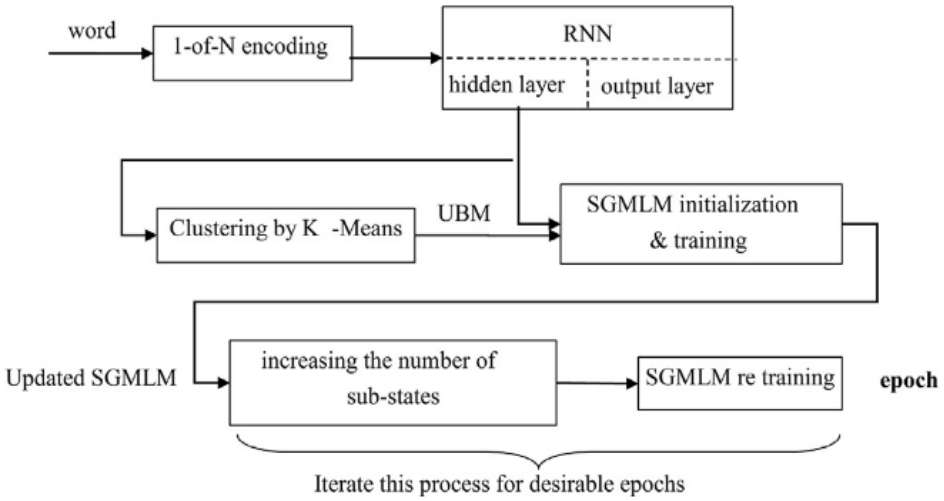

Figure 6. Block diagram for building SGMLM (Sun and Chol, 2020).

The model is trained and evaluated with several corpora and achieves very promising results for ASR systems. The SGMLMs architecture based on the Top-Down approach yields a WER of 5.7%, and the SGMLM architecture based on the Bottom-Up approach yields a WER of 6.01%.

Passricha and Aggarwal (2019) have proposed a hybrid system based on a deep convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) into a single model. This model folds the CNN, BLSTM and fully connected layers into a single system as is shown in Figure 7.

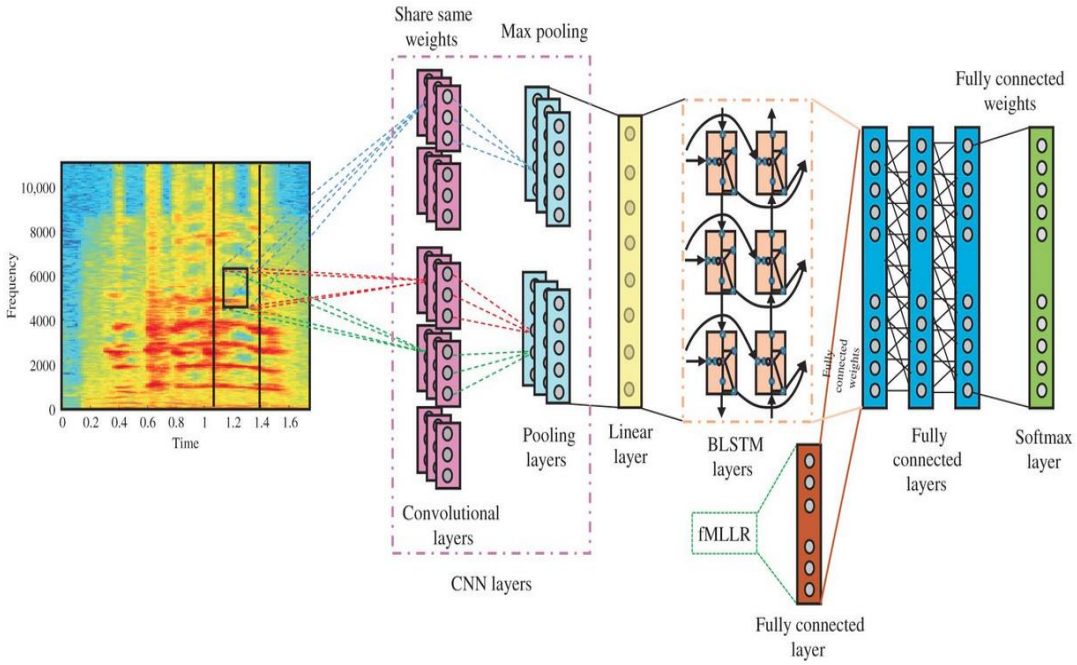

Figure 7. The CNN-BLSTM architecture (Passricha and Aggarwal, 2019).

The first module of this architecture is CNN, which is composed of some convolutional layers aimed to stabilize the variations that occur in the input signal. Because the feature dimension for speech is small, initially only two CNN layers are taken into account, which has 256 feature maps for each layer. The output of CNN layers goes as input to the pooling layers, and the pooling function is applied only for the frequency domain. The pooling layers summarize the features generated by convolution layers by reducing the complexity of the network. The features generated by the pooling layers feed the BLSTM network which aims to model them in time. The output of BLSTM layers feeds to fully connected layers, which generate higher-order feature representation into different classes. To solve the problem of inconsistency between training data and testing data it is applied the fMLLR-based language model adaptation. The output of the fMLLR layer also feeds the fully connected layer. The last module is the softmax function which translates the results into a normalized probability distribution.

This framework inherits the advantages of CNN, BLSTM, and fully connected layers, significantly improving the performance of the system by achieving a WER of 19.7% on the testing set.

Luscher et al (2019) have proposed another hybrid ASR system based on DNN-HMM. In addition, they have proposed an attention-based system to compare the performance of both models to each other.

Both models use bi-directional LSTMs for acoustic modelling (AM), and for language modelling (LM) is applied both LSTM and Transformers-based architectures.

For the hybrid ASR system, the AM is a bi-directional LSTM as described by Graves et al (2013). To feed the bi-directional LSTM are used Gammatone filters (Schluter et al., 2007) along with the generated alignments from the GMM/HMM model. In total there are six bi-directional LSTM layers with 1000 units for both backward and forward directions. In the output layer, a softmax layer is implemented, which together with the output unit, corresponds to the number of the CART labels. To train the network, are applied both the frame-wise cross-entropy loss (Kingma and Ba, 2014) and Nadam rules (Dozat, 2015). And to control the learning rate and prevent overfitting applied the Newbob learning rate rule (Zeyer et al., 2017).

Three language models (LM) were applied. The first model is a 4-gram-based language model as described by Kneser and Ney (1995). The second model is LSTM LM which is composed of two recurrent layers with 4096 LSTM nodes in each layer, an input layer with a size of 128, and an output softmax layer as described by Sundermeyer et al (2012). And the third model is a combination of LSTM with Transformers. The Transformers perform the rescoring of lattices generated by the LSTM LM. It is composed of 96 layers with self-attention, with a total dimension of 512, 8 heads and with an internal dimension of 2048 in each layer.

The second proposed system is based on attention-based encoder-decoder architecture. It operates on sub-word units through byte-pair encoding as described by Sennrich et al (2016). As input is used the MFCC features. To change the size of the encoder and the LSTMs is used a pre-training function, where initially the encoder has 2 layers with dimension 512 and then it is increased into 6 layers with dimension 1024. Also, in this system, two language models are applied. The first model is based on LSTM, while the second model is based on Transformers. The LSTM LM is composed of 4 recurrent layers with 2048 LSTM nodes. While the Transformers LM is composed of 24 layers, 8-head self-attention and a feed-forward with sizes 1024 and 4096.

Both systems are trained on training data by the LibriSpeech corpus. The encoder-decoder-attention architecture by combining with the Transformers LM yields a WER of 3.2% on test-clean and 9.9% on

test-other. While the hybrid system when rescoring with a Transformers LM yields a WER of 2.3% on test-clean and 5.0% on test-other, by outperforming the encoder-decoder-attention architecture.

Wang et al (2020) have proposed Transformers-based acoustic models (AMs) for hybrid ASR. They explore different coding methods but the main focus remains on the investigation of Transformers-based AMs without any constraint. The encoder converts the input sequences into embedding vectors, which aim to produce a senone or chenone for each frame, which is combined with lexicons and language models (LM) to construct a search graph. As an encoder can be used various neural networks such as DNN, TDNN, RNN, CNN, and their approaches. On the other hand, the decoder is used to find the best probability distribution or hypothesis. Figure 8 shows the architecture of one Transformers layer.

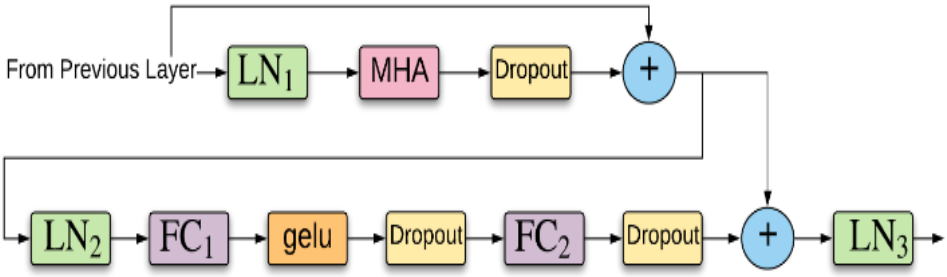

Figure 8. The architecture of one Transformers layer (Wang et al., 2020).

Each transformers layer is composed of the MHA sub-layer and a fully-connected feed-forward network (FFN). The FFN consists of two linear transformations and a nonlinear activation function between them. In addition, a normalization layer is applied before MHA and FFN as well as another LN is applied to connect with the Transformers. Another node in this layer is the GELU mechanism (Radford et al., 2018) which is applied in the FFN network.

The first step in this approach is to calculate the probability distribution of the input sequence through the self-attention mechanism. And to encode the positional features at the input of the Transformers three ways are addressed.

I. Sinusoid positional embedding aims to encode absolute positional information.
II. Frame stacking aims to encode the relative positional information

III. Convolutional embedding aims to conduct the collection of the input features and to learn the right models.

In the baseline architecture of Transformers, the RNN in the encoder and decoder is replaced by self-attention and cross-attention sequentially. While in the proposed hybrid system, self-attention replaces the RNN only in the acoustic encoder. And, to improve the overall model performance, it applies the self-attention mechanism and FFN networks integrated with deep approaches. In addition, the convolution layer is used as a pre-processing for the input of the Transformers layer.

The proposed architecture is evaluated based on the LibriSpeech corpus. All experiments that were done used context-position-dependent graphemes. Also, it is applied 1-state HMM with fixed self-loop and forward transition probability. It is applied to an 80-dimensional log Mel-filter bank to extract features. In addition, the speed perturbation and Spec Augment techniques are applied. The training of the model is based on PyTorch-based fairseq as described by Myle et al (2019). An Adam optimizer is applied in all experiments conducted. In addition to the other features that we have mentioned, one of the main innovations that this paper has brought is the application of 4-gram LM and 4-gram NNLM (Kneser and Ney, 1995).

Referring to the experimental results, when the model uses the standard 4-gram LM, it achieves a WER of 2.6% on the test clean and 5.59% on the test-others. While, when the model uses 4-gram + NNLM, it achieves a WER of 2.26% on test-clean and 4.85% on test-others.

Another hybrid system based on Transformers is presented by Pan et al (2020). This system brings two innovations in terms of architecture. The building of the acoustic model (AM) is based on multi-stream CNN architecture and the building of the language model (LM) is based on a self-attentive simple recurrent unit (SRU) architecture.

The multistream architecture stabilizes various temporal resolutions in multiple streams to improve the stability of the system. For various temporal resolutions, it considers stream-specific dilation rates on TDNN-F as described by Povey et al (2018). Applying the Spec Augment in multistream CNNs architecture makes the system more robust.

This architecture is composed of five 2D-CNNs layers for each stream aimed to process Mel spectrograms. In addition, each stream applies seventeen TDNN-F layers, where each layer applies a convolution matrix, followed by a skip connection, batch normalization and dropout layer. The output of this architecture is connected to the ReLu non-linear activation function, batch normalization layer and a dropout layer.

The language model (LM) is based on SRU architecture as described by (Lei et al., 2018). It applies the element-wise hidden-to-hidden connections, to provide their independence and parallel execution, which accelerate the training time significantly. To increase context modelling capacity is applied a self-attentive approach, where the linear operation is replaced by a multi-head attention operation (Vaswani et al., 2017).

This architecture is integrated with multiple phases of LM rescoring, where in the last phase is applied a self-attentive SRU LM interpolated with previous self-attentive SRU where one is trained through word pieces and the other word level. The interpolation enables the ranking of the N-best hypotheses generated by lattices rescored in the TDNN-LSTM LM.

The proposed models are trained and evaluated with the LibriSpeech corpus. They apply the RAdam optimizer, which uses a cosine annealing learning rate schedule. In addition, single-headed attention in the self-attentive node is applied.

Referring to experimental results, the best proposed LM is a 24-layer self-attentive SRU LM, which yields a WER of 1.75% on test-clean and 4.46% on test-other.

Zhang et al (2020) have proposed a hybrid ASR, which applies word-pieces as modelling units combined with CTC training. By integrating this approach with a Transformers network trained with chenone-CTC, great results are achieved by overcoming the existing models. The CTC applies a blank label mechanism to train the network by evaluating the alignment between the input sequence and the targeted sequence.

In this architecture, the encoder, which consists of 24 layers, is designed based on Transformers. Unlike baseline architecture, this model applies three VGG layers (Abdel-Hamid et al., 2014) to encode the acoustic features before they feed the Transformers layers. Each VGG block consists of two successive

convolutional layers, followed by a nonlinear ReLu function and a pooling layer. The max-pooling layer is applied to each VGG block with a stride of 1 to 3 for each block. To put a balance between recognition latency and accuracy in the proposed architecture the LC-BLSTM network is applied. The LC-BLSTM uses a limited number of right context frames to make predictions keeping delays under control. Each LC-BLSTM layer is composed of one left LSTM and one right LSTM. This approach generates high-dimensional audio features at the same time as the encoder is processing the audio sequence. Also, this architecture applied an iterated loss function (Tjandra et al., 2020), a layer normalization before multi-head attention (MHA) and a feed-forward network (FFN). In addition, a layer normalization after the residual connection prevents bypassing of the Transformers and accelerates the training time of the model. To model the parameters, the technique described by (Wang et al., 2019) is followed. The encoder generates a probability distribution of all labels. The word-piece unit divides words into a limited set of sub-word units as described by Wu et al (2016). This road-piece approach is modelled using word-piece vocab and blank labels, and it is trained by applying unigram LM as described by Kudo (2018). Innovation in this architecture is the application of chenone in CTC training. After the CTC model is trained, the decoding graph is built, first by following the standard procedure for CD-HMM and then it is transformed into a graph which uses a blank label.

The training and evaluation of the model were done using the LibriSpeech corpus. Two external language models have been applied: 4 grams LM and NNLM. Referring to the experimental results in the case where 4 grams of LM is applied, the model achieves a WER of 3.31% in test-clean and 4.79 in test-others. While, in the case where NNLM is applied, it achieves a WER of 2.1% in test-clean and 4.2% in test others.

Zeineldeen et al (2022) have proposed a conformer-based hybrid ASR system. The novelty of this architecture is the application of the time-down-sampling approach for model training as well as the application of transposed convolutions to remodel the desired output. Figure 9 shows the proposed conformer-based architecture.

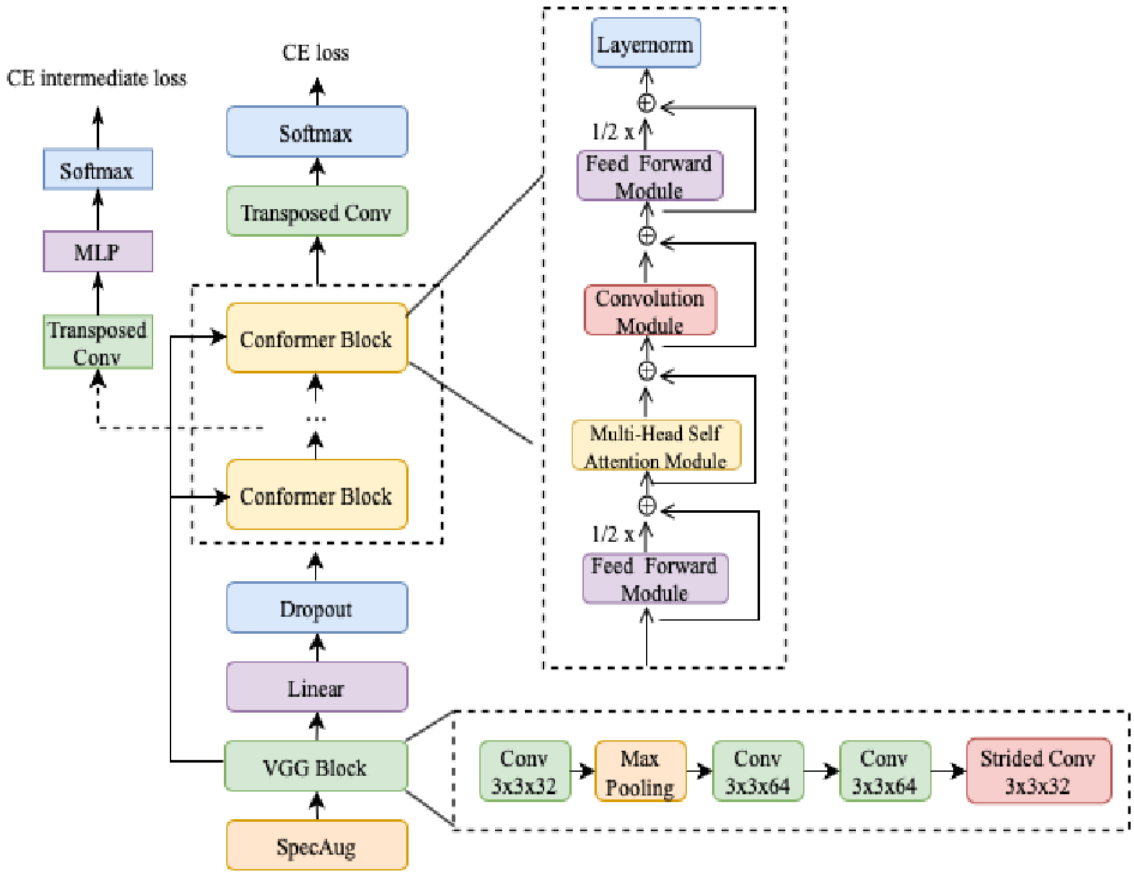

Figure 9. The proposed conformer-based architecture (Mohammand et al., 2022).

This system is composed of four main modules which are derived by baseline conformer architecture (Gulati et al., 2020): 1) feed-forward (FFN); 2) multi-head self-attention (MHSA); 3) convolution (Conv), and 4) feed-forward module. As can be shown in Figure 9 the NN is composed of several blocks. The first block is a VGG network as described by (Simonyan and Zisserman, 2015). Then, 4 convolution layers with 3×3 kernel size are applied. Between these layers, the swish activation function is applied, and between the first and second layers is applied to the max-pooling layer. While the fourth layer is used for time down sampling and for time up sampling one transposed convolution is used. These layers are followed by 12 conformer blocks. Each MHSA module has a size of 512 with 8 attention heads. Also in this architecture, a relative positional encoding is applied. The training of the model is based on frame-wise cross-entropy (CE) rules which are generated by the HMM-GMM system as described by Tuske et al (2015). Also, the Adam optimizer with Nadam (Dozat, 2016) as well as the Newbob learning schedule (Zeyer et al., 2017) is applied to control the learning rate. In addition, in all conformer modules the

dropout, embedding and attention are applied. While in the transposed convolution layers are applied the weight decay (Krogh and Hertz, 1991), which improves generalization. And to overcome overfitting the focal loss (Lin et al., 2017) is applied.

Time down and up-sampling methods are applied to train the model. For downsampling is used a stridden convolution is integrated into the VGG network. Also, a transposed convolution layer is applied to remodel the desired output. To train deeper networks, auxiliary losses in different layers are used, which affect the stability of the training (Wang et al., 2020). To reduce the model size the parameter sharing technique is applied. To reuse the learned features, the LongSkip connection is applied, which connects the output of the VGG network to the input of each conformer block. A focal loss function (Lin et al., 2017) has also been applied that restricts the CE objectives. In addition, the application of the Spec Augment technique (Park et al., 2019) masks out blocks in the time domain and frequency domain. And to reduce the combination between training and recognition is used Sequence discriminative training (Vesely et al., 2013). As LM is used the 4-gram count-based LM, LSTM LM and a Transformers-based LM for rescoring.

Referring to the experimental results, the proposed hybrid-based-conformer model, which applies Transformers-based LM, gives the best results, reaching a WER of 9.7% on the testing set. When it applies 4-gram LM it yields a WER of 11.4%, and when applying LSTM-LM 10.1% on the testing set. The consulted papers in identifying potential advanced hybrid speech recognition systems have been included in Table 1. As can be shown in Table 1, the topics include a combination of GMM-HMM and DNN-HMM models with different approaches to neural networks. It is also noted that nowadays, commercial hybrid systems have begun to be integrated with Recurrent Neural Network (RNN) and their different variants as well as with advanced techniques based on Transformers and Conformer.

Table 1. Analysis of advanced hybrid speech recognition architectures.

Authors	GMM-HMM	DNN-HMM	CD-GMM-	CD-DNN-	TDNN	DBN-ANN	SGM-HMM	SGMLM-RNN	CNN-RNN	LSTM-BLSTM	Transformers	Conformer
Dahl et al (2012)		√		√								
Jinyu Li et al (2012)		√		√								
Jaitly et al (2012)		√				√						
Kadyan and Kaur (2020)	√						√					
Sun and Chol (2020)	√							√				
Passricha et al (2019)									√	√		
Vegesna et al (2017)	√	√										
Luscher et al (2019)		√								√	√	
Wang et al (2020)		√								√	√	
Pan et al (2020)					√				√		√	
Zeineldeen et al (2022)										√	√	√
Zhang et al (2020)									√		√	

Referring to the analysis that we have done for each paper, each architecture has its advantages and disadvantages. But the focus of our study at this stage is focused on the overall performance of the system, referring to the standard performance measurement parameter word error rate (WER). In Figure 10, we graphically present the value of WER for each architecture.

First, we emphasize that even though WER is the basic parameter that determines the performance of the model, many factors directly affect the value of WER such as corpus size, corpus accuracy, speaker attributes, etc. The training and evaluation of the models analyzed in this session were not done with the same corpus. Despite this, the results presented in Figure 10, clearly show which architectures have the best performance.

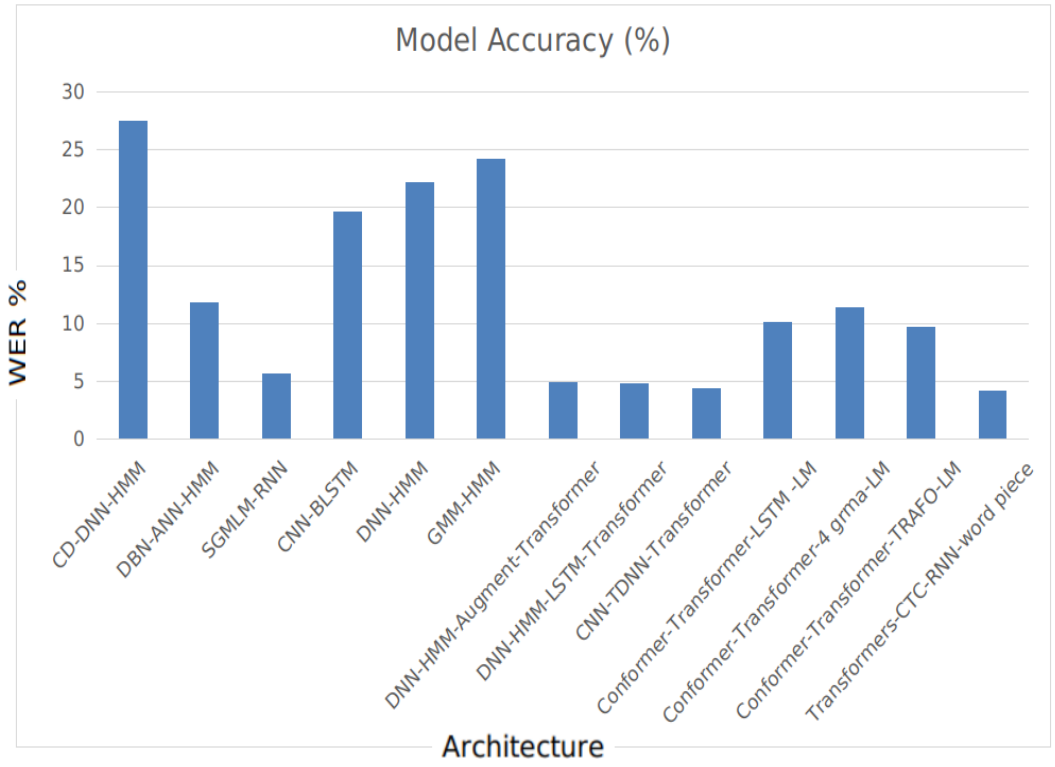

Figure 10. The WER values for all hybrid architectures were analyzed.

Referring to the results presented in Figure 10, we show that the architectures based on the commercial models HMM, GMM and DNN yield relatively good results. The hybrid architectures that combine various variants of RNN with Transformers achieve great accuracy. Also, we notice that the application of different language models reduces the WER by 2-5%.

2.4 End-to-end Speech Recognition Systems

End-to-end (E2E) SR systems are an emerging paradigm in the domain of speech recognition that offers multiple benefits (Yu and Deng, 2016). Unlike hybrid systems, the E2E ASR systems fold the acoustic model, language model and pronunciation model into a single neural network. This guarantees a simple training process, reduces the training time and decoding time, as well as improves the overall performance of the model. Today, end-to-end models being integrated with deep learning approaches have significantly improved the accuracy of ASR. The success of deep learning in speech recognition started with the presentation of fully-connected deep neural networks (DNN) (Sainath et al., 2015).

Recent advances in deep learning techniques applied to speech recognition include the usage of convolutional deep neural networks (CNN) (Mohammed and Al-Zawi, 2017) and the recurrent neural networks (RNN) versions (Iba and Noman, 2020). Applying deep learning in end-to-end ASR allows the model to learn directly from the data instead of the usage of feature engineering (Song and Cai, 2015). And today, the application of the Transformers in ASR systems has made a turn in the domain of speech recognition (Vaswani et al., 2017). Integration of Transformers with end-to-end approaches increases the accuracy of the model drastically, as well as speeds up the training time.

In this session, we will analyze the most cited and popular end-to-end ASR systems during the last years. For each selected architecture, we will make a detailed analysis of it, describing all the components with which it is built. In addition, we will analyze the performance of each architecture by referring to the international standard word error rate (WER) parameter. At the end of this session, we aim to define the best end-to-end architecture, which we can use as a good reference to design ASR systems for the Albanian language. The most popular and the most cited end-to-end model is Deep Speech presented by Hannun et al (2014). This framework is based on an optimized RNN-based training approach, and new features synthesis approaches to provide a huge amount of different data for training. This architecture, unlike traditional systems, uses a function that is robust against noise, reverberation, or speaker variation found in the corpus. The main block of this framework is an RNN approach which applies some GPUs and new partitioning techniques to enable parallel data processing. The structure of the proposed RNN approach is presented in Figure 11.

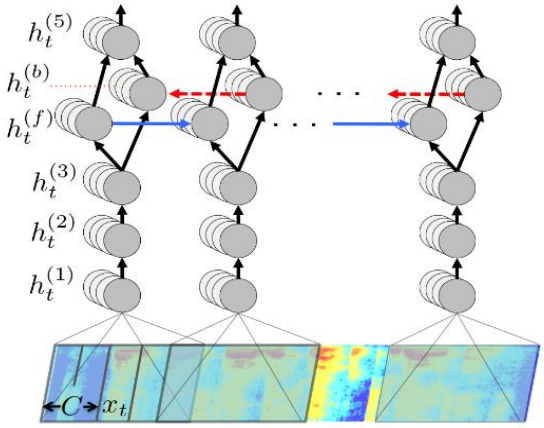

Figure 11. Structure of RNN model (Hannun et al. 2014).

The RNN consists of 5 layers of hidden units, which aims to convert the input sequence into a sequence of character probabilities. The first RNN layer is not recurrent, and at each time the output(y_t) depends on the speech spectrograms x_t along with a context of C frames on each side as can be shown in Figure 11. Also, layer 2 and layer 3 are non-recurrent layers and operate on independent data for each time step. The first three layers for each time t are computed:

$$h_t^{(l)} = g(W^{(l)} h_t^{(l-1)} + b^{(l)})$$

Where $g(z) = \min\{\max\{0, z\}, 20\}$ represent the clipped rectified-linear (ReLu) activation function and $W^{(l)}, b^{(l)}$ presents the layer feature matrix.

The fourth layer represents a bi-directional recurrent layer, which consists of a set with forwarding recurrence, and a set with backward recurrence. Mathematically it is expressed as:

$$h_t^{(f)} = g(W^{(4)} h_t^{(3)} + W_r^{(f)} h_{t-1}^{(f)} + b^{(4)}$$

$$h_t^{(b)} = g(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t+1}^{(b)} + b^{(4)}$$

The fifth layer, also is not recurrent and takes both the forward and backward units as inputs and outputs a softmax function that yields the predicted character probabilities for each time t and character k in the alphabet, which is expressed as:

$$h_{t,k}^{(6)} = y_{t,k} \equiv P(c_t = k \mid x) = \frac{\exp\left(W_k^{(6)} h_t^{(5)} + b_k^{(6)}\right)}{\sum_j \exp\left(W_j^{(6)} h_t^{(5)} + b_j^{(6)}\right)}$$

Evaluation of the prediction error is done through computation of the CTC loss function. While for the training of RNN it is used Nesterov's accelerated gradient technique as described by Sutskever et al., (2013). During the construction of the language model, the N-gram language model is used, as they are simpler for training with huge data and avoid more occurrences of errors during training [Coates et al., 2013]. This framework yields a WER of 16% on a full testing set.

Amodei et al (2016) have proposed a new improved version of Deep Speech architecture called Deep Speech 2. This model is based on the recurrent neural networks (RNN) approach, which applies 1D or 2D invariant convolutional input layers, followed by Vanilla or GRU layers and uni or bi-directional RNN

layers. Then it applies a look-ahead convolution layer followed by a fully connected output layer. The architecture of this model is shown in detail in Figure 12.

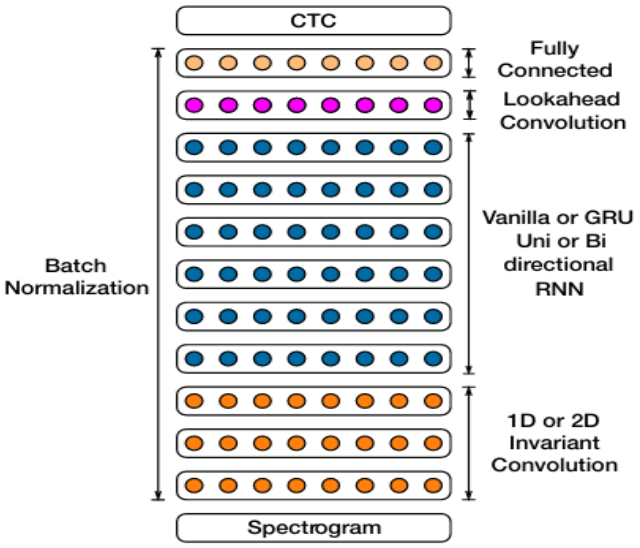

Figure 12. The architecture of Deep Speech RNN (Amodei et al., 2016).

The model is fed by the log spectrograms and outputs the characters. To simplify the training process in depth RNN is used the Batch Normalization (Batch Norm) technique as described by Ioffe et al., (2015). This technique called "sequence-wise normalization" consists of batch normalization only for the vertical connections. And for each hidden unit, the average and variance are calculated for the entire length of the sequences. In addition, the CTC loss function is applied to directly predict the characters and to make the training more stable. As shown in Figure 12, both GRU and Vanilla RNN mechanisms can be applied, which both support the Batch Norm and present strong performance. Another innovation in this architecture is the addition of a special layer called look-ahead convolution. This layer learns weights to linearly combine each neuron's activities into the future and provides control of the amount for future context. It can be placed above all RNN layers, by providing to transmit all computation below the look-ahead convolution in a detailed form.

Also, three invariant convolutional layers have been added to this architecture as shown in Figure 12. These layers are both in the time and frequency domain (2D) and only in the time domain (1D).

Application of the three 2D convolutions layers drastically improves the WER on the noisy training set. This architecture presents very good results, achieving a WER of 3.1% on the clean testing set.

Another very popular end-to-end model and one of the most cited is Listen Attend and Spell presented by Chan et al (2015). This model is based on sequence-to-sequence computation. The two main modules of this model are:

I. Listener represents an RNN-based encoder, which aims to convert low-level speech data features into higher-level features.
II. Speller represents an RNN-based decoder, which aims to convert the higher-level features into a probability distribution using an attention mechanism.

Both the encoder and the decoder are trained jointly. This approach can generate multiple spelling variants by overcoming the difficulties faced by the CTC model as a result of independence assumptions between frames. Figure 13 shows in detail the workflow.

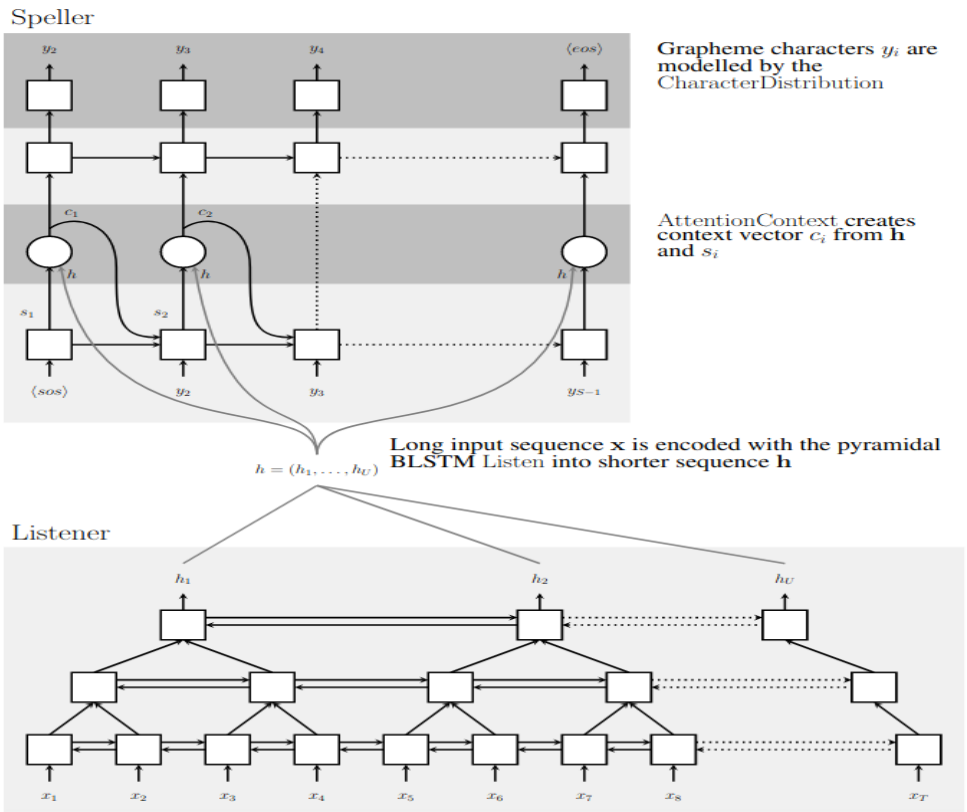

Figure 13. Listen, Attend and Spell (LAS) model (Chan et al., 2015).

The Listen (encoder) uses a pyramid BLST network (pBLSTMs), and as can be shown in Figure 13, it applies three of pBLSTMs on top of the bottom BLSTM layer. This modification overcomes the BLSTM performance, producing better results and converging faster by reducing computation complexity.

The Attend and Spell function is computed using an attention-based LSTM transducer as described by Chorowski et al (2015). The output of the transducer is a probability distribution for the next character which is produced taking into consideration all the previous characters. The attention mechanism processes the relevant acoustic features and outputs a context vector which represents the next character. This model presents great performance achieving a WER of 16% on full testing set without language modelling. In addition, it significantly reduces the training time and also reduces the complexity of the system as a whole.

Another much-cited model that achieves exceptional performance is proposed by Bahadanau et al (2016). This model is built based on the Recurrent Sequence Generator (ARSG) as a part of an end-to-end LVCSR system, applying an attention mechanism as described by Chorowsi et al., (2015). This model is based on the Recurrent Neural Network (RNN) which maps sequences of speech signals to sequences of characters. The RNN is used to build both the encoder and decoder, which deal with the variable length of the input and output sequences. The encoder is a deep BiRNN that aims to transform the input into a sequence of BiRNN state vectors. While the decoder integrates one RNN with an attention mechanism into an Attention-based Recurrent Sequence Generator (ARSG) aims to learn the alignment between input and output.

The novelty of this approach is precisely the application of an Attention-based Recurrent Sequence Generator (ARSG) as a part of an end-to-end LVCSR system. Figure 14 shows in detail the structure of the Attention-based Recurrent Sequence Generator (ARSG).

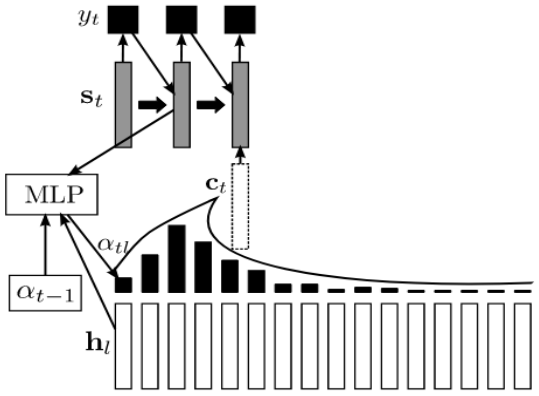

Figure 14. The Attention-based Recurrent Sequence Generator architecture (Bahdanau et al., 2016).

Unlike RNN, the ARSG outputs a sequence $(y_1 \ldots\ldots\ldots\ldots . y_{T)}$ by aligning simultaneously each generated element to the encoded input sequence $(h_1 \ldots\ldots\ldots\ldots . h_L)$. It consists of an RNN integrated into an attention mechanism which aims to update the hidden state of the RNN and to make a prediction for the next output sequence, based on the input sequence. The attention mechanism presented is an improved version of the attention mechanism described by Chorowski et al (2015), where a window is applied in the training phase. A modification that was made in the attention mechanism is its limitation in the positions $(m_{t-1} - w_l \ldots\ldots . . m_{t-1} + w_r)$, where m_{t-1} is the median of α_{t-1}, which means a distribution. The values w_l and w_r indicate the expansion of the window to the left and the right directions. This modification makes the training process converge extremely faster and also, makes ARSG training easier for longer input sequences.

This model achieves a WER of 18.6% without a language model, and it yields a WER of 9.3% when it integrates an extended trigram LM into the decoding process.

Zhang et al (2016) have proposed a very deep convolutional network for end-to-end ASR. This model is constructed based on four operations: 1) batch normalization; 2) ResCNN convolutional; 3) LSTMs and, 4) network in network logic. To clearly explain how this architecture works, we are briefly explaining each of its components and the contribution each of them has to this architecture.

I. Batch Normalization (BN) aims to normalize each input layer, speeding up the training process and improving the performance of the system as a whole.

II. Residual Networks (ResNets) aim to learn a residual function of the input using skip connection. The ResNets application enables the training of deep networks even without the use of complex optimization algorithms, as a result of the construction of a deeper acoustic encoder through a skip connection mechanism.

III. Convolutional LSTM (ConvLSTM) consists of a convolutional structure in both the input-to-state and state-to-state transitions which aims to replace the inner products within the LSTM unit as described by Shi et al., (2015). This enables the increase of computational power in the model by further reducing the number of its parameters to improve performance.

IV. Network in network (NiN) increases the depth of the network between LSTM layers. One NiN 1X1 or more convolution modules can be added to these LSTMs layers depending on the depth required. This allows for increasing the depth of a network and simultaneously increasing its predictive power, reducing the number of its parameters.

Integration of these four components jointly builds a very deep recurrent and convolutional structure, which exploits the spectral structure in the feature space and adds computational power deep into the network without causing overfitting. Figure 15 shows the architecture of the model in detail.

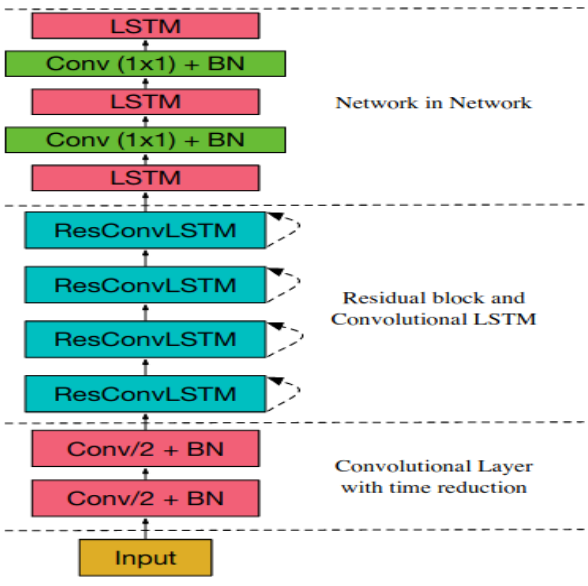

Figure 15. The architecture of the model (Zhang et al., 2016).

As is shown in Figure 15, this architecture is built by two convolutional layers at the bottom, then followed by four residual blocks and LSTM NiN blocks. Each residual block contains a convolutional LSTM as well as a convolutional layer. As can be seen in Figure 15, the ResConv module is replaced by the ResConvLSTM module. The model is trained and evaluated with the WSJ corpus. It achieves good results, yielding a WER of 10.5% on the testing set without language modelling.

Chung-Cheng Chiu et al (2018) have proposed some optimization techniques to overcome the performance of the baseline Listen to Attend and Speller (LAS) architecture. First, the authors have proposed to modify the structure of the model by applying the word piece models (WPM) as described by Chan et al (2017). WPM is an approach which segments a word into multiple subwords. Each subword represents a token. Second, they have applied a multi-head attention module (MHA), which allows the model to jointly attend to different representation features from different locations as is shown in Figure 16.

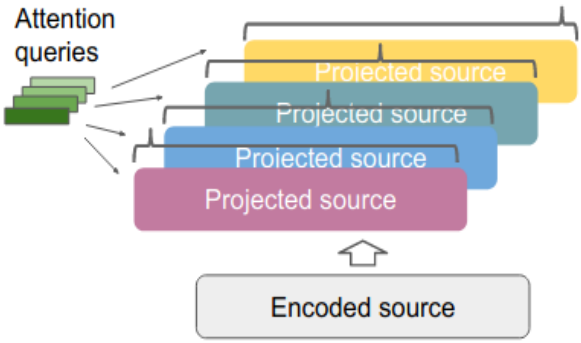

Figure 16. Multi-head attention module (Chiu et al., 2018).

Each head has a specific function and is designed to attend to the encoder output, by greatly simplifying the encoder-decoder communication. Third, the authors have proposed to optimize the process, which includes four strategies:

I. Minimization of the expected number of word errors (MWER) (Prabhavalkar et al., 2018).

II. Application of scheduled sampling (SS) to feed the previous label prediction during the training process (Bengio et al., 2015).

III. Application of label smoothing regularization mechanism, which improves prediction (Szegedy et al., 2016).

IV. Training of the model with a synchronous training mechanism (Goyal et al., 2017).

The model is trained and evaluated using a data set of 12,500 hours in the English language derived from the Google database. Referring to the experimental results, these modifications greatly improve the performance of the system, yielding a WER of 4.1 % on its training test.

Park et al (2019) have proposed an improved version of Listen Attend and Spell (LAS), applying a Spec Augment technique to the baseline LAS architecture. This technique is applied directly to the feature inputs of RNN, operating on the log Mel spectrogram of the input audio wave and does not require any additional data. Spec Augment is composed of three types of deformations on the log Mel spectrogram, which helps the network learn useful features.

I. Time warping consists of the deformation of the time series in the time direction.

II. Time masking masks a block of consecutive time steps.

III. Frequency masking masks a block on consecutive frequency steps.

Application of Spec Augment technique on LAS model greatly improves its performance. It yields a WER of 6.8% on its testing set of LibriSpeech corpus without the use of a language model.

Li et al (2019) have proposed an end-to-end convolutional neural network acoustic model (CNN-AM) called Jasper. This model has replaced the AM and PM models with CNN. It processes the input Mel-filter bank features and produces a probability distribution for each frame. The baseline architecture is presented in Figure 17.

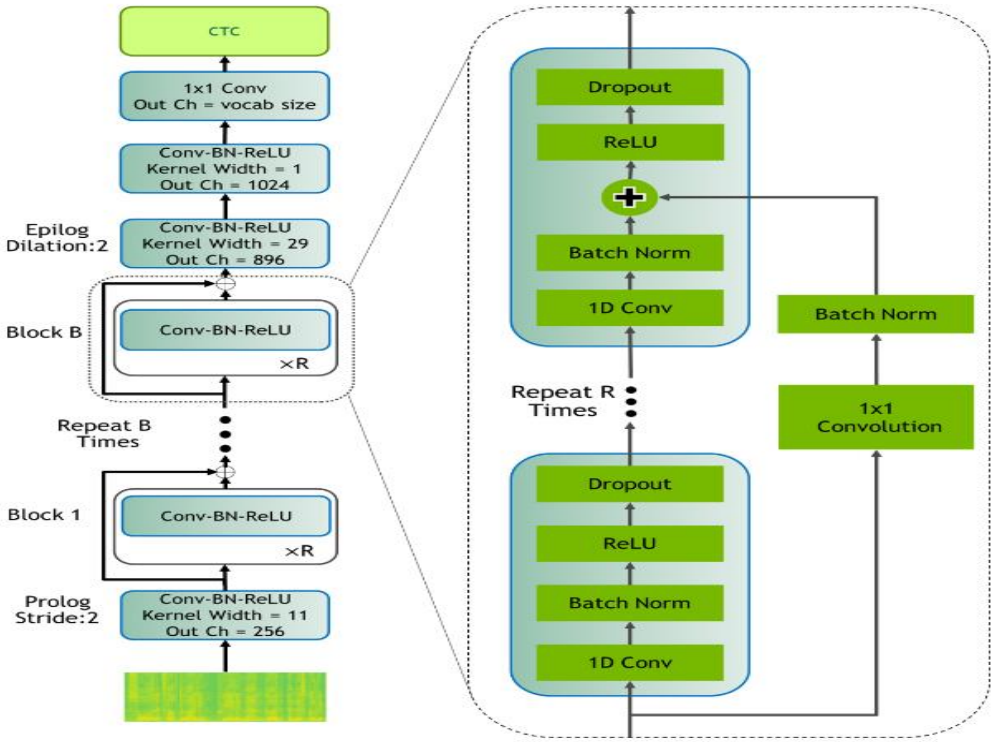

Figure 17. The Baseline Jasper architecture (Li et al., 2019).

It is composed of B blocks, where each block has R sub-blocks. In each of the sub-blocks are applied four operations: 1) a 1D convolution; 2) batch norm; 3) ReLU; and 4) dropout.

The input of each block is connected directly with the last sub-block through a 1x1 residual connection, followed by a batch norm layer. Then the sum of all the blocks passes to an activation function and a dropout layer, which produce the final output. All these operations are presented clearly in detail in Figure 17. The structure of this architecture drastically increases the training time of the model using the GPU quite well. The authors have also proposed another variant of the basic Jasper architecture called Jasper Dense Residual (DR). In this architecture different from the basic Jasper architecture the output of a convolution block is added to the inputs of all the subsequent blocks as described by Huang et al., (2016) and Tang et al., (2018). The evaluation of the model was done with two corpora: LibriSpeech and Wall Street Journal (WSJ), and in both cases very good results were obtained. Using LiriSpeech, a WER of 3.86% was obtained without a language model. While, when the model is trained with WJS, it achieves a WER of 13.3% without a language model.

Another model in this domain is presented by Li et al (2019). This model represents an improved version of the baseline Recurrent Neural Network-Transducer (RNN-T) architecture [He et al., 2019]. The improvements made in this paper consist of two directions:

I. Changes in the architecture of the model.

The changes in the architecture consist of the implementation of the layer trajectory that separates the classification tasks and temporal modelling tasks using both depth-LSTM and time-LSTM and both depth-GRU and time-GRU approaches. Figure 18 presents the concept of a layer trajectory diagram.

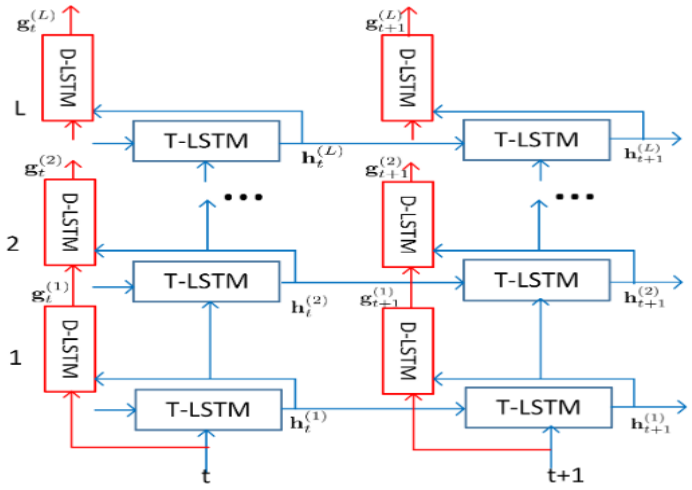

Figure 18. Diagram of layer trajectory LSTM (ltLSTM) (Li et al., 2019).

The Depth-LSTM (D-LSTM) module in this architecture is applied to examine the features generated by time-LSTM (T-LSTM) modules to all layers within the current time slot to classify senones. As can be shown in Figure 18, the time recurrence occurs only in the T-LSTM module, while in the D-LSTM module it does not happen. Some other improvements proposed in this context are:

a) Replacement of the depth-LSTM formulation which works across layers for the target classification with the look-ahead embedding vector.

b) Application of gated recurrent unit (GRU) as a building block for RNN-T to define the layer normalized GRU.

c) Proposal of layer trajectory GRU (ltGRU) with layer normalization to perform temporal modelling and target classification.

d) Expansion of the GRU trajectory layer (ltGRU) using the look-ahead embedding vector.

II. Optimizations in the training algorithm.

One of the challenges of the model based on RNN-T is memory consumption during the training process. For this purpose, this architecture aims to optimize the RNN-T approach by integrating the encoder with the predicted features, as well as reorganising the gradient features by preventing the storage of tensors in memory. Referring to the experimental results, this architecture shows great performance with all the proposed improvements in comparison with the basic RNN-T architecture. It reduces the WER up to 6.6%.

Zhang et al (2020) have proposed an end-to-end speech recognition model with Transformers encoders. The novelty of this model is the replacement of the RNN and RNN-T encoder with Transformers encoders. This enables the training of the model with a self-attention mechanism, significantly reducing the training time as well as the performance of the model. Figure 19 presents in detail the architecture of the Transformers encoder.

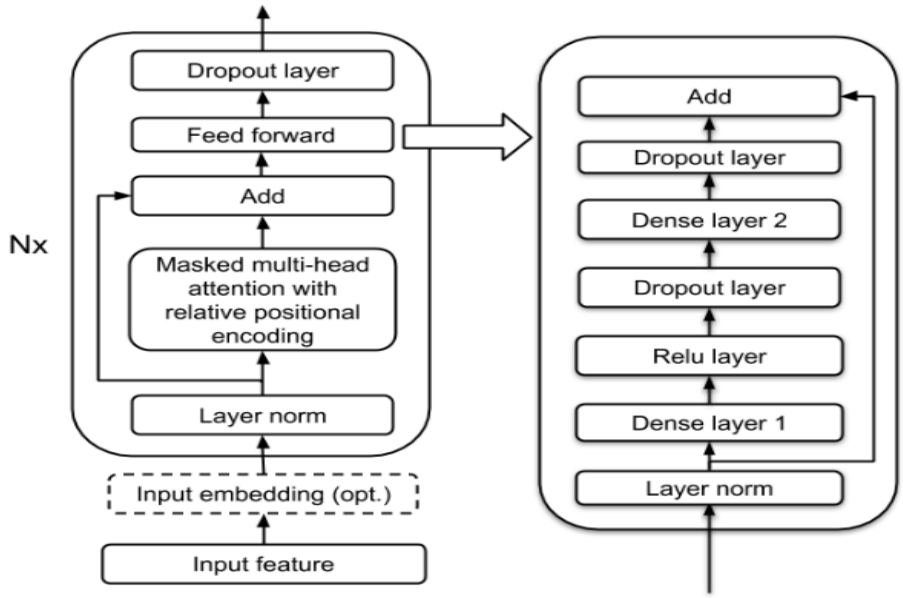

Figure 19. Transformers Encoder architecture (Zhang et al., 2015).

The encoder is composed of multiple uniform layers, multi-headed attention and a feed-forward layer. The multi-headed attention layer aims to first apply the Layer Norm and after that design the input to Query, Key, and Value for all the heads. Each attention head applies the attention mechanism independently by efficiently controlling the context of the model. The outputs of all heads are collected together and passed to a dense layer. Then a residual connection is applied to extract the final output of the multi-head attention sub-layer. Another module in this architecture is the dropout mechanism, which is applied to the output of the dense layer to avoid overfitting. The feed-forward sub-layer first applies the Layer Norm and two dense layers. Also, a ReLu activation function is used in the first dense layers.

Then, the dropout module is applied in both dense layers for regularization as well as a residual connection of normalized input and the output of the second dense layer as is shown in Figure 19. In addition, a relative positional encoding mechanism is applied as described by Dai et al., (2019). This mechanism ensures that the model follows a sequential order and affects only the attention score. This mechanism significantly reduces the complexity of the model. The model is trained and evaluated with the LibriSpeech corpus, and achieves a WER of 4.2% on the testing set, by outperforming all end-to-end architectures built with RNNs and their versions.

Gabriel et al (2020) have proposed some improvements to semi-supervised learning-based architectures. These architectures are based on end-to-end models and consist of improvements to acoustic models and language models. First, we will present 3 acoustic models proposed by the authors, explaining their innovations. And then we will present the proposed language models.

I. The ResNet Acoustic Model consists of some 1-D convolutions blocks, which apply skip connection to reduce the vanishing gradient problems in DNNs. The encoder is composed of forty-two convolutional layers, some of which are placed between ResNet blocks to increase the sampling rate of the number of hidden layers. In each convolutional layer is applied the ReLU, dropout and Layer Norm by increasing the depth of the network. In addition, three max-pooling layers are placed along the network. This architecture applies both the CTC and Seq2Seq loss functions.

II. Time-Depth Separable (TDS) Convolution AM architecture is based on the TDS block as described by Hannun et al (2019). It consists of a 2-D convolution layer, two fully-connected layers followed by a ReLU and Layer Norm as well as the residual connections. But in this architecture, an expansion of the TDS block is proposed, increasing the number of internal fully-connected layers to increase the size of the model. To guarantee an optimal context size for the encoder 3 1-D convolution layers with stride 2 and kernel size of 21 × 1 are applied. The training is done only in the context of labelled data, where three TDN blocks are used. To avoid overfitting problems is applied dropout in all TDS blocks.

III. Transformers-Based AM is composed of 3 1-D LibriSpeech AM convolutions layers or 6 1-D LibriVox AM convolutional layers. Each layer has a kernel width of 3 and applies a GLU activation function. Each Transformers block is composed of four attention heads, an FFN and a nonlinear ReLu. Both CTC and Seq2Seq models are applied. In the CTC model, the output of the encoder is followed by a linear layer, while in the seq2seq model is added a 256 decoder size which folds 6 Transformers and 4 attention heads.

Regarding language models, authors have considered three variants: n-gram LM, convolutional GCNN LMs and Transformers-based LMs. For n-gram LM, and GCNN LM is applied to both word-based and word-piece models, while Transformers-based LM is applied only to word-based models. In both the word-piece and word-level GCNN LM models, are applied the GCNN-14B architecture as described by Dauphin et al (2017). While in the word-level Transformers LM is applied the same architecture as described by Baevski et al (2019).

As a decoder is used the lexicon-based and lexicon-free beam-search decoders are integrated with a LM as described by Likhomanenko et al (2019). To train and evaluate the proposed architectures the LibriSpeech corpus is used. Referring to the experimental results the best acoustic models are based on Transformers and yield 6.98% WER without decoding and 5.71% WER with decoding and scoring on test-others. While the AMs Transformers-based with both CTC and Seq2Seq yield a WERs of 4.88% without decoding on test-other and yield a WER of 2.28% on the test clean. And when the decoding and scoring are applied it yields a WER of 2.09 on the test clean and 4.11 on the test-others.

Gulati et al 2020 have proposed a new end-to-end architecture called Conformer, which combines the advantage of Transformers to model global features and the advantage of CNN to model local features of an audio sequence. Figure 20 presents in detail this architecture. The encoder is composed of a convolution module that reduces data size, followed by a linear and a dropout layer and at the output some conformer blocks. A conformer block consists of 4 main modules: 1) a self-attention module; 2) a feed-forward module; 3) a convolution module and, 4) a feed-forward module in the output. The convolution module is composed of a gating mechanism, a point-wise convolution and a gated linear unit (GLU), followed by a 1-D depth-wise convolution layer and a batch-norm function aimed to train deep architectures. A very important module of this architecture is multi-headed self-attention. It is integrated with a positional encoding architecture derived from Transformers XL (Dahi et al., 2019). This technique enables the self-attention mechanism to adjust for different lengths of the audio sequence as well as to increase the stability of the encoder against the utterance length. To train and stabilize the parameters of the model, a prenorm residual unit with dropout is applied as described by (Wang et al., 2019).

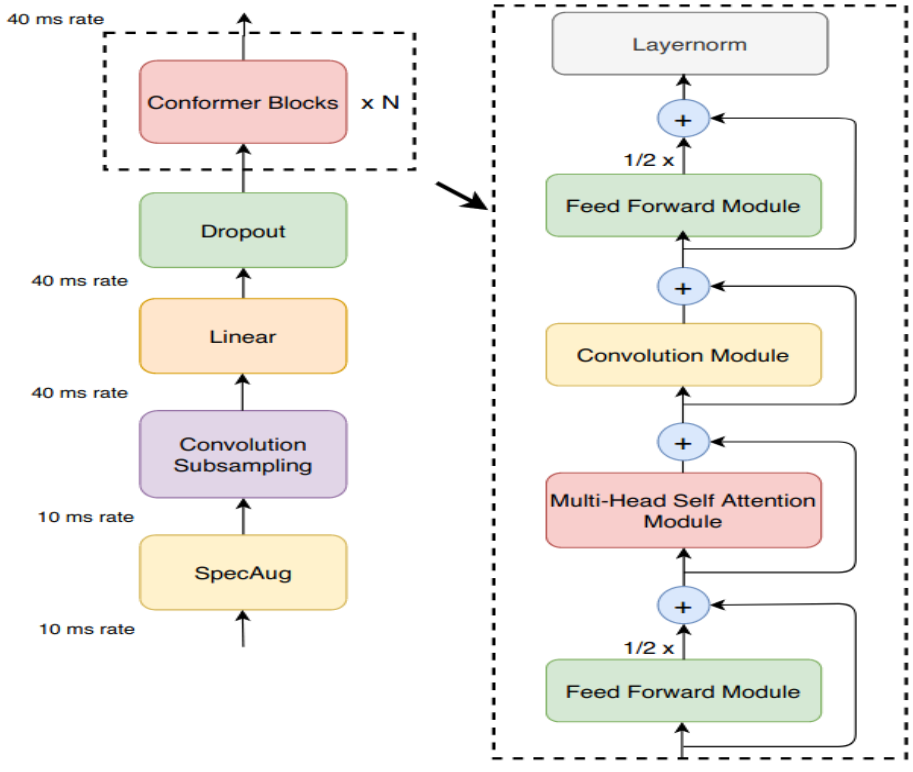

Figure 20. The Conformer architecture (Gulati et al., 2020).

The Transformers deploy a prenorm residual unit (Dahi et al., 2019) and applies the layer normalization inside the residual unit as well as on the feed of the first layer. Also, the Swish activation function (Ramachandranet et al., 2017) and dropout are applied to make the network more robust. The proposed architecture is based on a sandwich structure scheme as described by (Lu et al., 2019). In the Transformers block, the feed-forward layer is replaced with two half-feed-forward layers, one before the attention and one after. The features generated by these half-layers are applied to feed-forward (FFN) modules. The training and evaluation of the proposed model were done with the LibriSpeech corpus. As the encoder is applied a single LSTM layer for all proposed models. In each residual unit is applied a dropout function aims to regularize the model. Also, a Spec Augment technique as described by Chan et al (2019) is applied to improve the accuracy of the model. Both architectures with and without language models (LM) were studied. As LM is applied a 3-layer LSTM LM. Referring to the experimental results, the Conformer architecture shows exceptional results in both cases with and without LM. And specifically, in the Conformer architecture without LM, the WER reaches 2.1% in the test-clean and 4.3% in the test-others. When applying LM in test-clean the WER reaches 1.9% and in test-others, it reaches 3.9%.

Kriman et al (2020) have proposed an end-to-end model for ASR based on the Jasper architecture (Li et al., 2019) called Quartz Net. The innovation that this architecture brings compared to the baseline Jasper architecture is the replacement of 1D convolution modules with 1D time-channel separable convolutions modules as well as the implementation of depthwise separable convolutions. The Quartz Net architecture starts with a 1D convolutional layer and is followed by multiple blocks with residual connections between them. Each block is composed of four layers: a normalization layer, K-sized depthwise convolutional layer, a pointwise convolution layer and a ReLU layer. And at the end, 3 convolution layers are applied. Figure 21 shows the Quartz Net architecture.

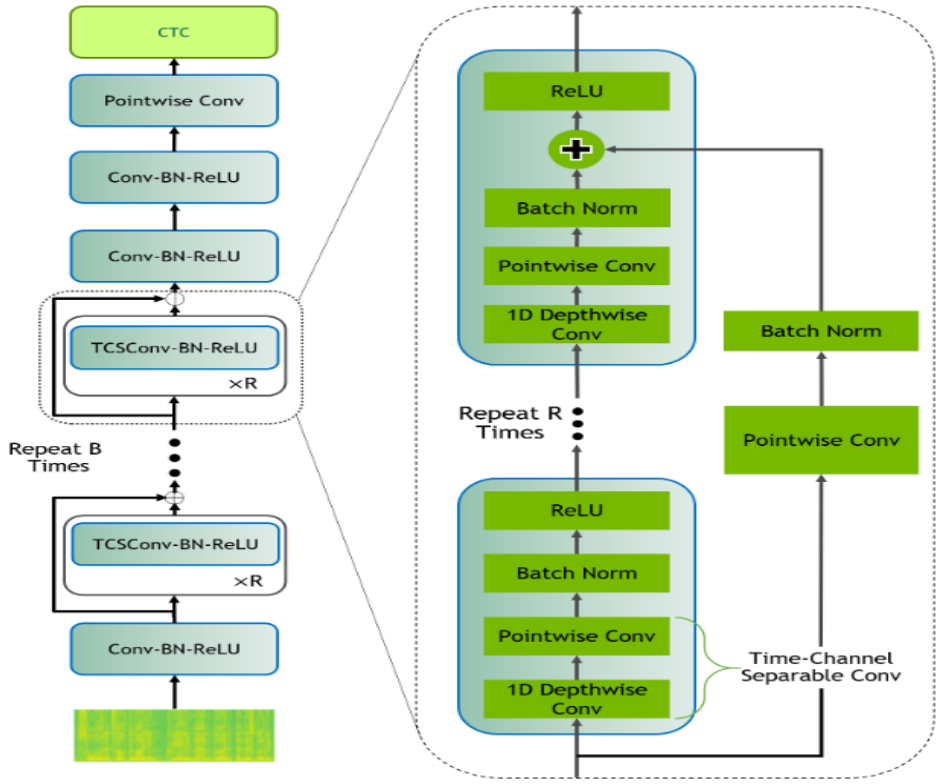

Figure 21. Quartznet architecture (Kriman et al., 2020).

The depthwise convolution is applied to each channel enabling the kernel to be widely used compared to the basic Jasper architecture. For more stable training and better performance, batch normalization has been applied (Ioffe et al., 2015). To reduce the number of model parameters, in pointwise convolution layers are applied group convolutions. In addition, a group shuffle layer is used to improve interchange between groups. This will significantly reduce the training time of the model. The model has trained with Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) and applies two external language models: 6-gram LM and Transformers-XL LM. To train and evaluate the proposed models the LibriSpeech corpus is used. And referring to the experimental results when 6-gram LM is applied it achieves a WER of 2.96% in the test-clean and 8.07% in the test-others. While, when applied Transformers XL LM, it achieves a WER of 2.69% in test-clean and 7.25 % in test-others. While in the case when no LM is applied, it achieves a WER of 3.9% in test-clean and 11.28% in test-others.

The consulted papers in identifying potential advanced end-to-end speech recognition systems have been included in Table 2.

Table 2: Analysis of advanced end-to-end speech recognition systems.

Authors	end-to-end	RNN	CNN	LAS	Spec-Augment	MHA	LM	BiRNN	ResCNN	RNN-T	Transformers	Conformer
Awni Hannun et al (2014)	√	√										
Amodei et al (2016)	√	√	√									
Chan et al (2015)	√	√		√								
Park et al (2019)	√			√	√							
Chiu et al (2018)	√			√		√						
Li et al (2019)	√		√				√					
Bahdanau et al (2016)	√	√						√				
Bahdanau et al (2016)	√	√					√	√				
Zhang et al (2016)	√		√						√			
Li et al (2019)	√	√								√		
Zhang et al (2020)	√										√	
Gabriel et al (2020)	√			√							√	
Gulati et al (2020)	√		√								√	√
Kriman et al (2020)	√										√	√

As can be shown in Table 2, the first end-to-end models were mainly focused on LAS approaches and different versions of neural networks such as RNN, CNN, BiRNN etc. It is noted that the CTC and Spec Augment approaches have been applied in most of the architectures. And in recent years, a trend has been observed toward end-to-end models based on Transformers which are integrated with different RNN approaches.

Referring to the analysis that we have done for each paper, each architecture has its advantages and disadvantages. But the aim of our study at this stage is focused on the overall performance of the system, referring to the standard performance measurement parameter word error rate (WER). In Figure 22, we graphically presented the value of WER for each architecture.

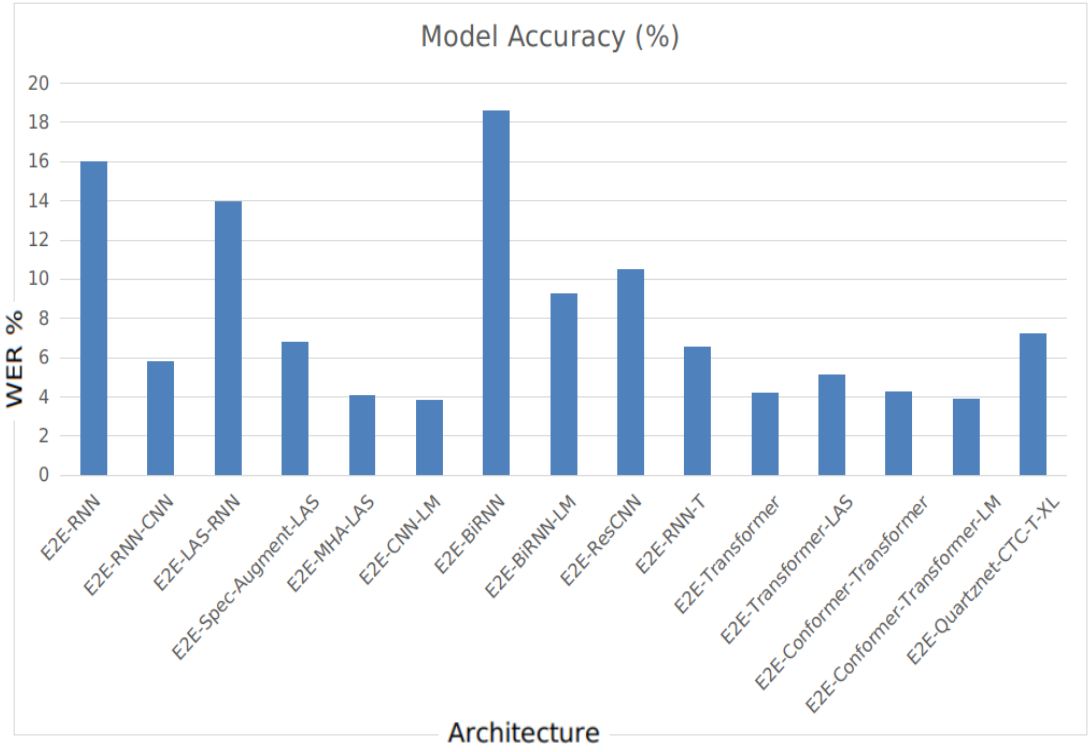

Figure 22. The WER values for all end-to-end architectures were analyzed.

First, we emphasize that not all analyzed architectures were trained and evaluated with the same corpus. Since corpus size, corpus accuracy, speaker attributes and other parameters are not the same, all these factors jointly affect the accuracy of the model. Despite this, the results presented in Figure 22 clearly show which architectures have the best performance. Referring to the results presented in Figure 22, we show that the end-to-end architectures based on RNN, CNN and various variants of them, have great accuracy. We also noticed that the application of the Spec Augment approach significantly increases the accuracy of the model. While end-to-end Transformers-based systems achieve state-of-the-art performance, overcoming the performance of end-to-end based RNN and their approaches systems. Also, we notice that the application of language models reduces the WER by 2-8% in some architectures.

2.5 ASR systems for low-resource languages

In this session, we will perform an analysis of the most popular and cited architectures in the domain of speech recognition, which have been applied in low-resource languages. Also, we will make a description

of the way of training and testing for each architecture, highlighting at the end of each architecture its performance.

Comparing architectures to each other is very complex since each language has its characteristics, including morphological, syntactic, grammatical or semantic specifics. Also, dialects, socio-linguistic specifics, ethnocultural specifics or even the physical specifics of the person who articulates the word make this process more complex. To analyze the performance of each architecture, we have referred to the international standard word error rate (WER) parameter. At the end of this session, we aim to define the best architecture, which we can use as a good reference to design ASR systems for the Albanian language.

Vegesna et al (2017) have proposed a model based on DNN-HMM for the Telugu low-resource language. In addition, they have compared the proposed DNN-HMM model with the baseline GMM-HMM hybrid model. Figure 23 presents the proposed architecture with all its modules.

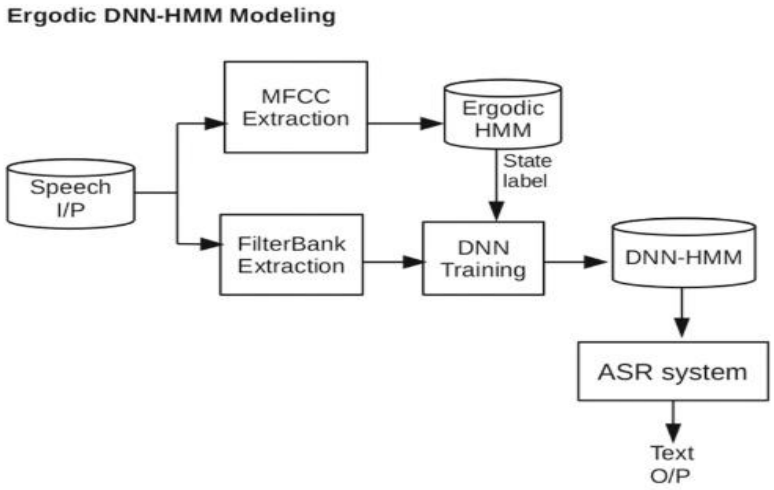

Figure 23. Ergodic DNN-HMM architecture (Vegesna et al., 2017).

The novelty of this architecture compared to the baseline DNN-HMM model (Dahl et al., 2011) lies in the addition of a node called ergodic HMM. In this node, the sequential features of the observation are modelled making them more suitable for classification. This classification consists of GMM-HMM training for all classes that have features like MFCC coefficients. The ergodic HMM is trained in the initial phase with MFCC features and uses the maximum likelihood estimation as described by Saul et al (2000), for

learning. To evaluate the probabilities of current states and transit states as well as the mean and covariance of GMMs are used Baum-Welch (Levinson et al., 1983) and EM algorithms (Digalakis et al., 1993). This architecture combines the strong learning power derived from DNNs and sequential modelling derived from HMMs. Referring to the experimental results, the DNN-HMM hybrid model achieves a WER of 22.2%, overcoming the performance of the GMM-HMM model, which achieves a WER of 26.2%.

Fathima et al (2018) have proposed a multilingual Time Delay Neural Network (TDNN) system for low-resource language. It combines multilingual acoustic modelling (AM) with language model (LM) to decode the input test sequences. Figure 24 shows in detail the architecture for both the training phase and the testing phase.

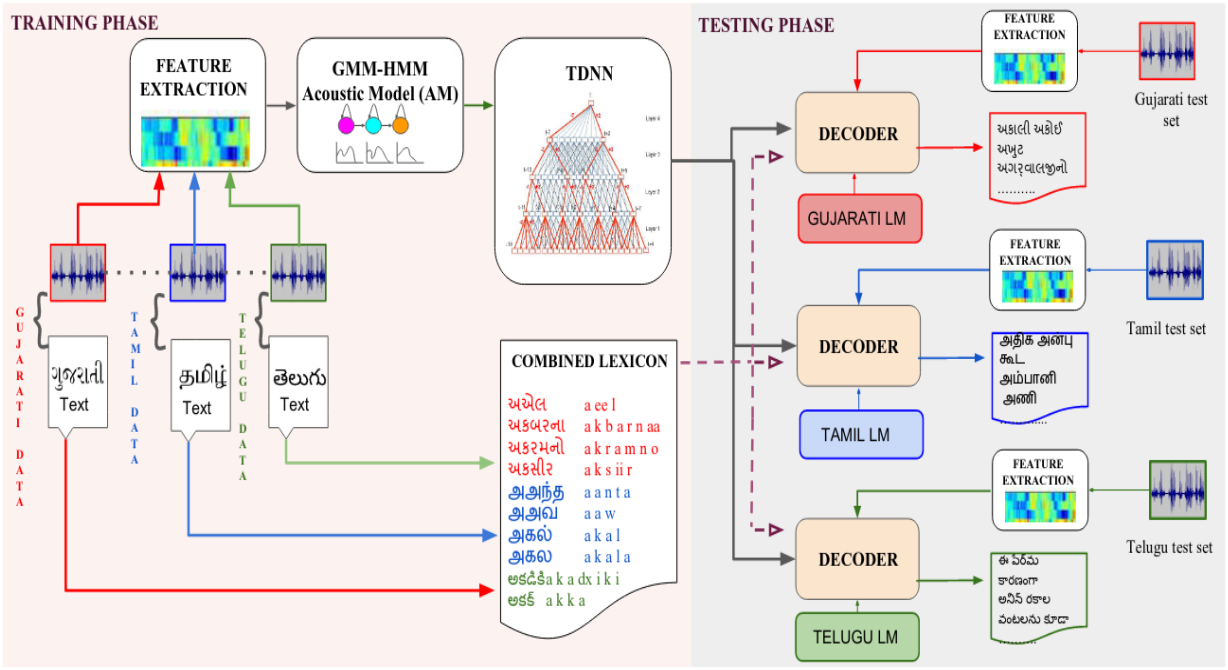

Figure 24. The architecture of the TDNN ASR system (Fathima et al., 2018).

The first step in the training phase is the features extraction process, which uses MFCC features without cepstral truncation. Then the GMM-HMM acoustic model based on FMLLR features is built. The next step consists of training the TDNN chain, where the alignments obtained from the acoustic model are used to feed it. Another module in this architecture called combined lexicon is used to cover the words that occur in the training set of all three languages. While in the testing phase features are extracted

from the test audio and a language-specific language model is used for decoding to obtain output transcriptions for each language. This model achieves great results, specifically for the Tamil language it yields a WER of 16.07%, for the Telugu language it yields a WER of 17.14% WER and for the Gujarati language, it yields a WER of 17.69%.

Zhou et al (2018) have proposed an end-to-end ASR system with a single Transformers for low-resource language. This model integrates all modules of an ASR system into a neural network following sequence-to-sequence attention-based rules. One of the innovations of this architecture is the processing of sub-words generated by byte pair encoding (BPE) (Sennrich et al., 2015) as a multilingual unit without the need for a multilingual modelling unit. Second, the usage of the Transformers as the basic module of the architecture solves the problem of training data that have languages with low resources. And specifically, trained Transformers from a high resource language are adopted as an initial model where the softmax layer is replaced by the language-specific softmax layer. Figure 25 shows the architecture of the ASR Transformers. This model is trained and evaluated with several high-resource languages, but it also is trained and evaluated with low-resource languages and it achieves an average of 12.4% of WER.

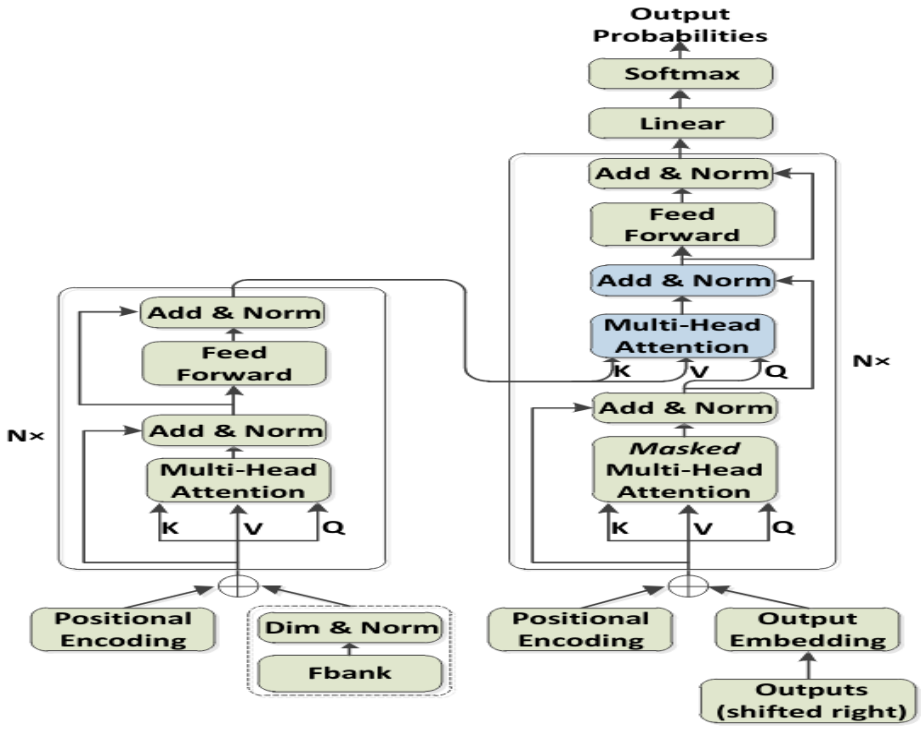

Figure 25. The architecture of the ASR Transformers (Zhou et al., 2018).

This architecture folds the multi-head attention (MHA), position-wise, and applies several fully connected layers in the encoder as well as the decoder. The encoder folds several uniform layers, where each of them consisting of two sub-layers. The first represents the MHA, while the second represents the position-wise FFN. The decoder is built in the same way as the encoder but with a difference, another sub-layer aims to conduct MHA across encoder output. In both encoder and decoder around every two sub-layers, residual connections and a normalization layer have been added. To regulate the flow of information in the decoder, these sub-layers mask out all values of the irregular connections. In addition, the positional encoding is added to the input at the bottoms of the encoder and decoder, to feed more information about the sequence, such as the position of the tokens. A linear transformation with a layer normalization is also added to this architecture, which converts the log Mel Filter-bank features into dimensions to combine the dimensions of the model, as shown in Figure 25 with a dotted line. The model is trained and evaluated with a dataset derived from the CALL HOME corpus which consists of 6 different languages.

Baevski et al (2019) have proposed a vq-wav2vec model to learn discrete speech representations through a wav2vec-style self-supervised learning context. This model combines the learning of discrete representations of speech sequences through context prediction (Dunbar et al., 2019) with learning speech representations in a self-supervised way through predicting context sequences (Steffen et al., 2019). This enables the application of different algorithms to speech data. The vq-wav2vec approach aims to learn discrete representations of fixed-length sequences of the speech signal by using the wav2vec loss and architecture as described by Schneider et al (2019). Two of the most important algorithms applied in this architecture are the Gumbel-Softmax algorithm as described by Jang et al (2016), and the k-means clustering algorithm as described by Eloff et al (2019). Both algorithms aim to select discrete variables. In addition, Deep Bidirectional Transformers are trained over the discretized unlabeled speech sequences to feed an acoustic model (AM) (Delvin et al., 2018).

The baseline wav2vec architecture as described by Schneider et al (2019) is composed of two convolutional neural networks where the encoder represents each time step which learns representations of the speech signal by solving self-supervised context-prediction tasks through a loss function as described by Oord et al (2013). While the vq-wav2vec architecture has two convolutional

networks like wav2vec, combined with a new quantization module to learn the VQ features of the speech signal.

The Gumbel-Softmax provides choices of discrete codebook variables by using the straight-through estimator. It applies two linear layers and a ReLU between them. One innovation that this architecture brings is the application of BERT pre-training where the task is to predict masked input tokens about the context (Delvin et al., 2018). The baseline BERT architecture has 12 layers, and 12 attention heads. The first step is BERT model training, which applies masked input token prediction (Liu et al., 2019). Then, it constructs the representations and feeds them into an acoustic model (AM).

The vq-wav2vec/wav2vec model is adapted with the fairseq implementation of wav2vec as described by Schneider et al (2019). As an acoustic model (AM) is applied the wav2letter as described by Collobert et al (2019). The proposed model has been trained and evaluated with several corpora, but we have analyzed the results only for the WSJ corpus. And referring to the experimental results, this architecture gives satisfactory results, reaching a WER of 4.46%.

Chen and Yang (2020) have proposed a framework for Yi low resource language. We describe both the training phase and the recognition phase. The training phase includes the training of the acoustic model (AM) and the language model (LM). While in the recognition stage, the Yi lexicon language is combined with the acoustic feature for joint recognition and decoding to output Yi transcription. The language model is built with the ternary LM by first analyzing the Yi text to obtain the acoustic features. It aims to produce the best text sequence in the speech recognition stage. As acoustic models, HMM, DNN, TDNN, and end-to-end AM were taken into consideration, which is trained independently.

In the GMM-HMM AM, the GMM estimates the probability distribution of the observation and uses the Baum-Welch algorithm to adapt this probability as the observation probability matrix of the HMM. While the HMM builds the speech sequence to obtain the state transition matrix.

The DNN AM training process is done by evaluating the posterior probability of the acoustic features extracted by the Yi corpus by replacing GMM. To train the DNN is applied an unsupervised training method which operates in each layer separately. During DNN training the input extracts acoustic features

and the output of the hidden layer is activated by the sigmoid function to feed the next layer, and in the last layer, the output is classified from a softmax function to predict the probability distribution.

The TDNN AM overcomes the problems that traditional GMM-HMM models have. In this case, the network does not need to align the phonetic symbols and speech audio in the timeline. The features of the hidden layer are related to the input of the current time, as well as the input of the future time. In this way, the TDNN can be connected and compare the past with the future. The TDNN reduces the number of weighted connections, classifies patterns with shift-invariance, and models context at each layer of the network.

The end-to-end folds the acoustic model, pronunciation dictionary and speech model into a neural network, significantly reducing its complexity. The core of the end-to-end in this paper is a CTC-attention model as described by Hori et al (2017). Figure 26 shows in detail the proposed framework of Yi speech recognition.

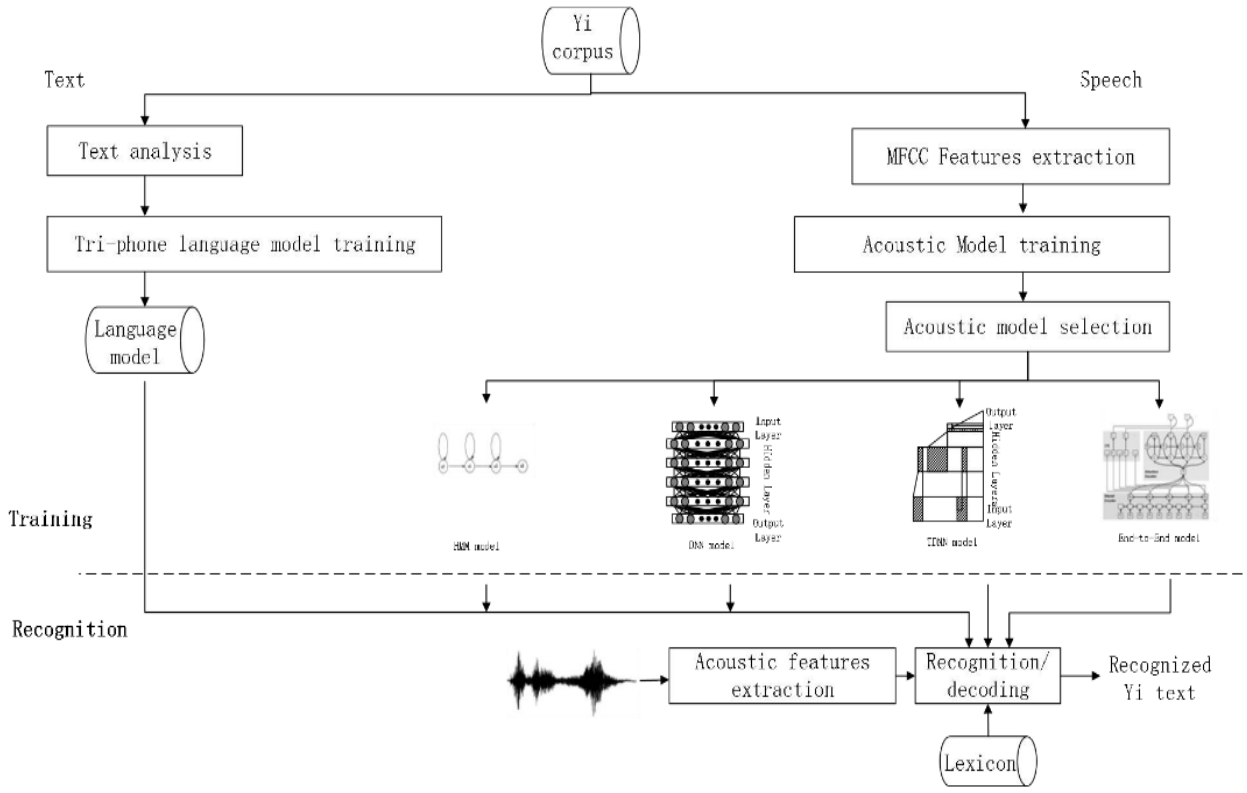

Figure 26. The framework of Yi speech recognition (Chen and Yang, 2020).

To solve the problem of low resource data, those low-resource languages have in general, it is used 3-gram LM of and regression smoothing technique. While the decoding process combines the 3-gram LM logic with a dictionary and AM.

The proposed models in this paper are trained with a mini corpus of the Yi language. And the evaluation of the model was done in the testing set. Referring to the experimental results, for the MFCC-GMM-HMM model a WER of 19.72% was obtained, for the MFCC-DNN model the WER reaches 16.72%, and for the MFCC-TDNN model the WER reaches 16.65% and for the MFCC end-to-end, WER achieves in 47.40%. We notice that the model based on TDNN AM achieves the best performance, while the end-to-end model achieves the worst performance, this is because these models require to be trained with larger corpora.

Wang et al (2020) have proposed an end-to-end low-resource SR architecture using the Tibetan language for training and testing. This model is based on the Deep CNN-LSTM network. The basic components of this architecture are: 1) acoustic feature extraction; 2) encoder; and 3) decoder. Figure 27 presents in detail the proposed architecture.

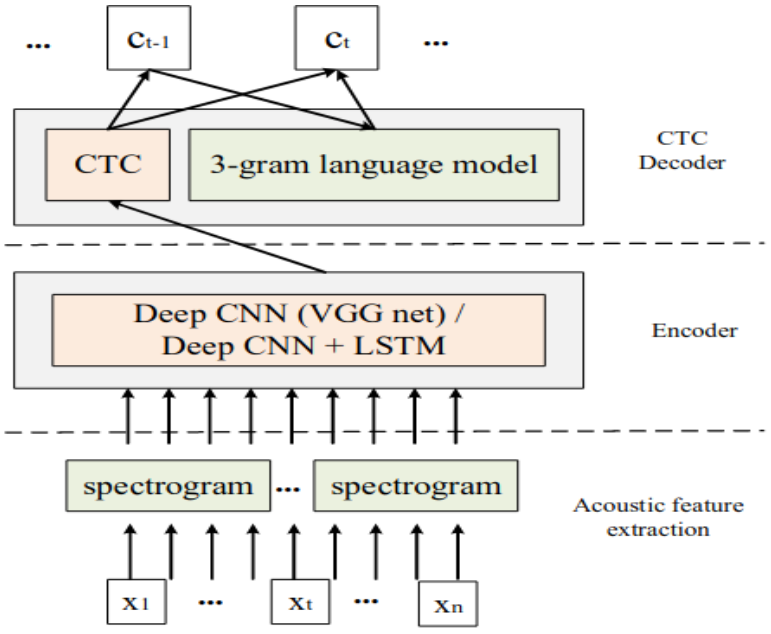

Figure 27. The architecture of low-resource SR (Wang et al., 2020).

The encoder is constructed by combining a single convolutional neural network (CNN) with a VGG network and long short-term memory (LSTM). Specifically, it consists of 10 CNN layers, a single LSTM layer with 512 memory blocks, which follows the last hidden layer of CNN. These layers are connected with an activation function with a learning rate of 0.008, while the output layer applies the softmax function for classification with a learning rate of 0.0001. To avoid overfitting 2 dropout layers are applied between the convolutional layers. A CTC decoder integrated with a 3-gram language model (LM) to decode the output of the encoder in the context of speech recognition is applied. This model is trained and evaluated using the Tibetan corpus that was built for this purpose. It yields a WER of 34.64%.

Anoop and Ramakrishnan (2021) have proposed an end-to-end model for Sanskrit low resource language. This model is based on CNN's and bidirectional GRUs networks as described by Amodei et al (2016) and Nguyen (2021). Because of the limitations of the low resources languages, a data augmentation technique called Spec Augment has been applied to increase the amount of data available for training (Park et al., 2019). As a feature extraction technique is applied the Mel- spectrogram which is supported by frequency masking and time masking techniques. To assess the performance of the learning CTC AM the greedy decoding algorithm (Zenkel et al., 2017) is applied and WFST is used to word level decoding (Mohri et al., 2022). Figure 28 presents the proposed architecture.

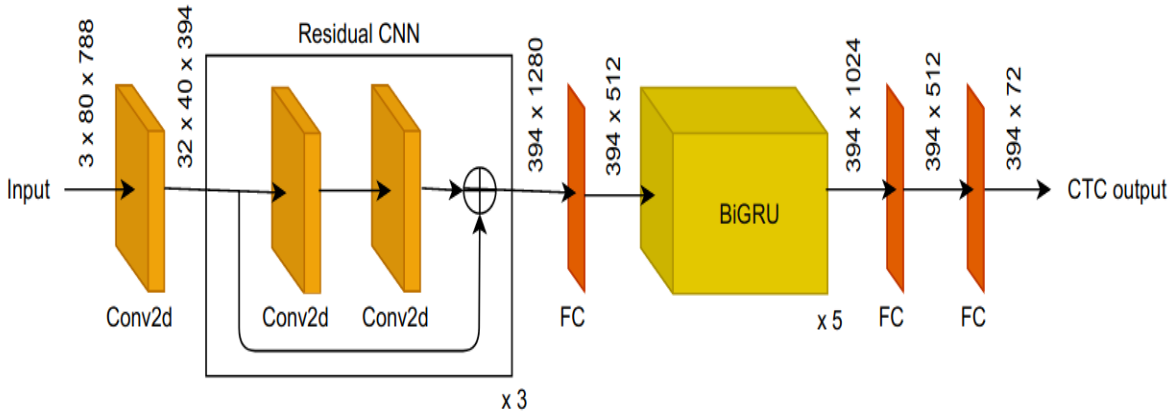

Figure 28. Block diagram of the proposed architecture (Anoop and Ramakrishnan, 2021).

Specifically, this architecture is built from 3 residual CNN blocks and 5 BiGRU blocks. Each layer in the CNN block uses a kernel size of 3X3, the stride of 1 and padding of 1, except the first layer which uses a stride of 2. In both CNNs and BiGRU is applied a layer normalization and an activation function called

Gaussian error linear units (GeLU). In addition, a dropout with a size of 0.1 is applied in each phase of the network.

The model is trained according to CTC policies by applying the Adam optimizer algorithm (Zhang, 2018). It presents quite satisfactory results in terms of performance by achieving a WER of 21.5%, and in the case when the Spec Augment technique is applied, the WER was drastically reduced, reaching 7.64%.

Meng et al (2021) have proposed a Mixspeech ASR system for low-resource language. MixSpeech aims to extract and split targets by mixed speech inputs to make the model applicable to other languages. It takes as input two different speech sequences that can be Mel-spectrograms or MFCC and output characters from both sequences. It uses each label to evaluate the recognition loss by combining the losses of both sequences. The novelty of Mixspeech is an augmentation of the speech input with another speech, which acts as a contrastive signal to improve the ASR performance. Authors have applied the MixSpeech in two end-to-end SR architectures: Listen, Attend and Spell (LAS) (Chan et al., 2015) and Transformers (Vaswani et al., 2017). In the LAS-based architecture, the encoder consists of four Bi-LSTM layers while the decoder consists of a LSTM layer. In the Transformers-based architecture, the encoder consists of twelve layers and the decoder consists of six layers. Both architectures apply the MHA and several fully connected layers. Also, they apply a CTC-attention mechanism to train the model. In the LAS architecture both the encoder and decoder are folded, the BiLSTM and LSTM respectively. While in Transformers they are folded of multi-head attention and feed-forward network. During the training process, it is applied to multi-task learning by combining CTC and Cross-Entropy.

The evaluation of the proposed techniques is done through three low-resource corpora: TIMIT, WSJ, and HKUST. Also, 3 different architectures are evaluated for both LAS and Transformers. The first case includes the baseline architecture, the second case in the basic architecture Spec Augment technique is applied and in the third case, the MixSpeech technique is applied. In all cases, satisfactory WER results were obtained, but when the model is trained with the WSJ corpus and the MixSpeech technique is applied, it achieves the best WER result of 4.7%.

Another architecture based on Transformers for low-resource language is presented by Xue et al (2021). The innovation of this architecture is the application of Bayesian learning for Transformers language modelling (LM). Figure 29 shows the architecture of the proposed model.

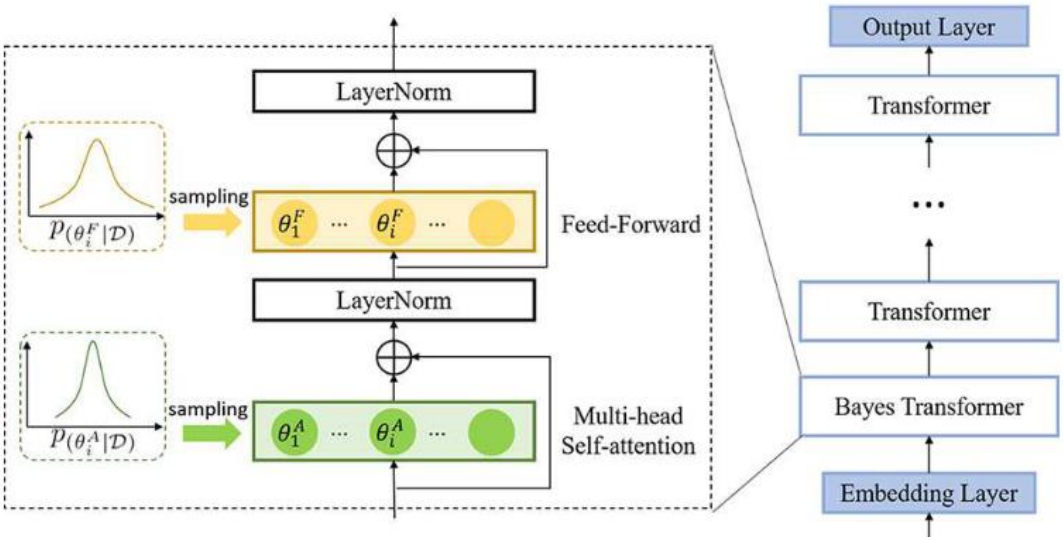

Figure 29. The proposed Bayesian Transformers language model architecture (Xue et al., 2021).

Unlike the baseline Transformers architecture [Vaswani et al., 2017], which is built by an encoder and a decoder, in this architecture, the decoder is adapted for language modelling as described by Li et al (2020). The decoder is composed of six blocks, where each of which has a MHA and a feed-forward module. Also between these modules the residual connections and layer normalization have been added as can be shown in Figure 29. Residual connection simplifies the training of the model by allowing gradients to flow through the networks directly without passing through non-linear activation functions. While layer normalization is applied to stabilize the network and reduce the training time. As the activation function is applied the Gaussian error linear unit (GELU) in the feed-forward module as described by Hendrycks and Gimpel (2016).

The novelty in this architecture is the application of the Bayesian Neural Language Model, which aims to model the parameter as a posterior probability distribution. This improves the accuracy of the model by increasing the prediction of unsafe words and also avoids overfitting and poor generalization. To train the Bayesian Transformers LMs is used a stochastic gradient and learning algorithm which makes a reparameterization of the variational lower bound. It gives a lower bound estimator that can be straightforwardly optimized using standard stochastic gradient methods as described by Kingma and Welling (2013).

This model is trained and evaluated using Switchboard and DementiaBank corpora, both of which are low-resource. After different architectures have been tested the Bayesian Transformers LM achieves the best performance yielding a WER of 7.6%.

Baevski et al (2021) have proposed an unsupervised speech recognition model based on wav2vec called wav2vec-U. It trains the SR models without any labelled data. This model uses a self-supervised approach as described by Baevski et al (2020c). This architecture has been applied to several languages, including low-resource languages. Figure 30 presents all steps of the proposed model.

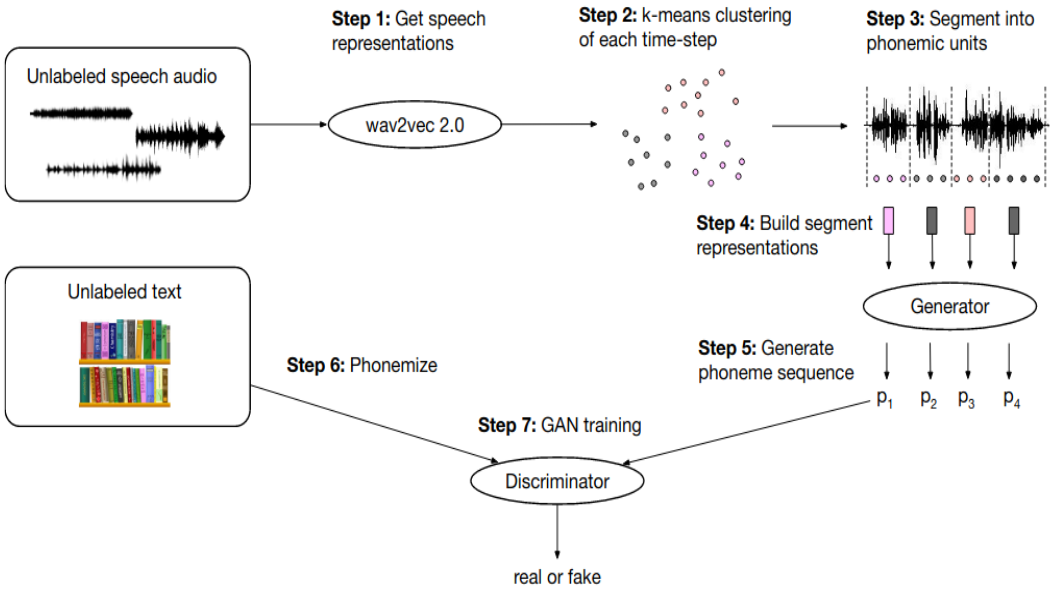

Figure 30. The wav2vec Unsupervised architecture (Baevski et al., 2021).

As can be shown in Figure 30, the workflow for this architecture goes through 7 steps.

I. Extraction of speech features using wav2vec 2.0 on unlabeled audio waves. The Wav2vec 2.0 is composed of an encoder which processes the audio waves and converts them to feature representations which are converted by Transformers into context representations. The Transformers is based on BERT architecture (Vaswani et al., 2017), wherein in the training process, the latent representations are discretized with a Gumbel softmax module to represent the targets in the objective (Jang et al., 2016).

II. Identification of clusters in the representations with k-means each time-steps.

III. Segmentation of the speech audio into phonemic units. This process is based on clustering wav2vec 2.0 speech representations through the k-means algorithm. Each cluster is labelled with a cluster ID, and every time this ID changes, the speech segment boundaries are incremented.
IV. Building of segment representations using a PCA and two mean pooling of the wav2vec 2.0 features. PCA retains only the most important features and generates an average representation of the segment.
V. Generation of phoneme sequence by the generator.
VI. Phonemize
VII. GAN training using the features of the unlabeled audio waves and the unlabeled phonemized text data.

Two very important components in this architecture are Generator and Discriminator. The generator is a single layer (CNN), which processes all the phonemes in each segment to generate the phonemes with the highest probability. It generates samples that are indistinguishable from the discriminator.

Also, the discriminator is represented by a CNN which generates a probability of the sample within the data set. It classifies the samples according to the generator or the real data distribution. Both, are trained by generative adversarial networks as described by Goodfellow et al (2014).

This architecture has been applied to several high-resource languages but it has also been applied to low-resource languages. The wav2vec-U with self-training (wav2vec-U + ST) and a Transformers LM yields a WER of 5.9% on the LibriSpeech testing set. While in the case when the same architecture is trained in the Swahili language, it achieves a WER of 32.2%. And in the case when this model is trained with Multilingual Librispeech (MLS) corpus, it achieves a WER of 18.6 %.

Pengcheng et al (2020) have proposed an end-to-end speech processing framework based on Conformer architecture. This framework is applied in many speech processing applications such as speech recognition (SR), text-to-speech (TTS), speech translations (ST) and speech separation (SS), but we will focus only on speech recognition applications. The proposed architecture has been tested for both low-resource and high-resource languages. The proposed architecture is composed of a Conformer encoder as described by (Gulati et al., 2020) and a Transformers decoder. In addition, Conformer folds a relative positional encoding approach by Transformers XL to generate better position information for the input

sequence with variable input length. The encoder is built from several blocks, where each block has integrated several position-wise feed-forward (FFN) modules, a convolution (CONV) module, and a multi-head self-attention (MHSA) module. The MHSA module aims to learn an alignment, where each token learns to gather from other tokens across one sequence.

The main module of this architecture is the CONV module. It starts with a 1-D pointwise convolution layer that aims to double the input channels followed by a gated linear unit (GLU) that aims to split the input along the channel and perform an element-wise product as well as a 1-D depth-wise convolution layer.

As can be shown in Figure 31, a Swish activation, a batch normalization layer, as well as one other 1-D point-wise convolution layer have also been applied in this architecture.

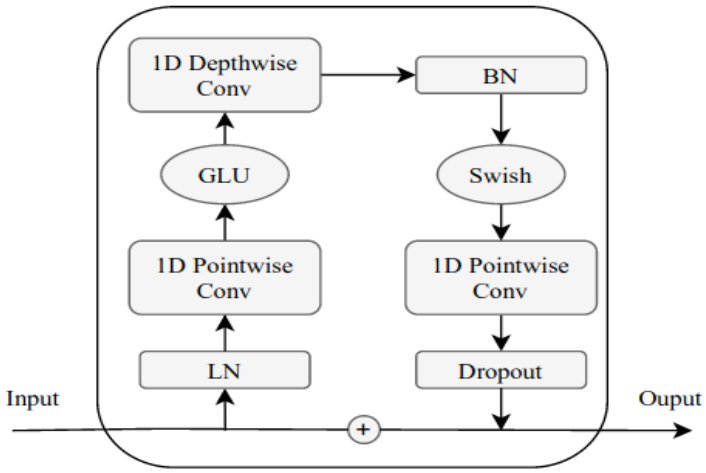

Figure 31. The architecture of the convolution module (Pengcheng et al., 2020).

Another important module in this system is the pointwise feed-forward module (FFN). It consists of two linear transformations with a ReLU activation function between them. In the Conformer architecture, two FFN modules integrated with the MHSA and CONV modules as well as the ReLUs are replaced with the Swish activation function.

The novelty of this architecture consists of the application of a pair of FNN modules, relative positional encoding and the integrated CONV module as described by (Lu et al., 2020). The training and evaluation of the proposed architecture were done with both low-resource and high-resource languages. In some

architectures, both speed perturbation and Spec Augment techniques have been applied. Model configuration parameters are referenced in ESPnet Transformers (Karita et al., 2019). Referring to the experimental results, where the model is trained with 8 different low-resource language corpora, in all cases the proposed architecture exceeds 15% WER compared to architectures based on Transformers. The best WER results are obtained when the data augmentation technique is applied, and specifically for the Persian language a WER of 2.1% was obtained and the average WER for all 8 languages is 12.8%.

Thienpondt and Demuynck have proposed an ASR architecture for low resource language, which folds data augmentation, transfer learning, Transformers, wav2vec 2.0 model and source-filter warping strategy into a single model. The first step toward the design of this architecture is the application of a data augmentation technique. And specifically, it is applied to a technique based on the source-filter model of speech as described by Fant (1981). It applies a VTLP function (Jaitly et al., 2013), which extracts the centre frequencies of the filter banks in the Mel-spectrogram representation. The second step is the application of transfer learning techniques, which adapt a model trained on a large corpus to perform robustly on a low-resource corpus. In this architecture is applied transfer learning on a Transformers model pre-trained on an adult corpus with a masking objective to build an ASR system for children as a low-resource scenario.

The proposed model is based on the XLS-R Transformers model (Babu et al., 2021), which is built over the wav2vec 2.0 approach (Baevski et al., 2020). It is composed of the encoder and the context networks. The encoder is composed of several blocks which include a GELU activation function (Gimpel et al., 2016), and several convolution blocks, which aim to convert the speech signal into a sequence of representations. The context network is composed of several Transformers blocks, and it aims to model contextualized acoustic representations. It is fed by speech representations.

First, the model is pre-trained with an adult speech corpus using a self-supervised technique and a loss function as described by (Babu et al., 2020). Then, in the network context, a linear layer has been added to produce the context representations of the ASR vocabulary.

The model is improved by applying the augment of adult speech with the source filter warping (SFW), which represents a data augmentation technique. In this way, the network is more robust against

children's speech. The model is optimized by applying the Connectionist Temporal Classification (CTC) function. Figure 32 shows the full proposed architecture.

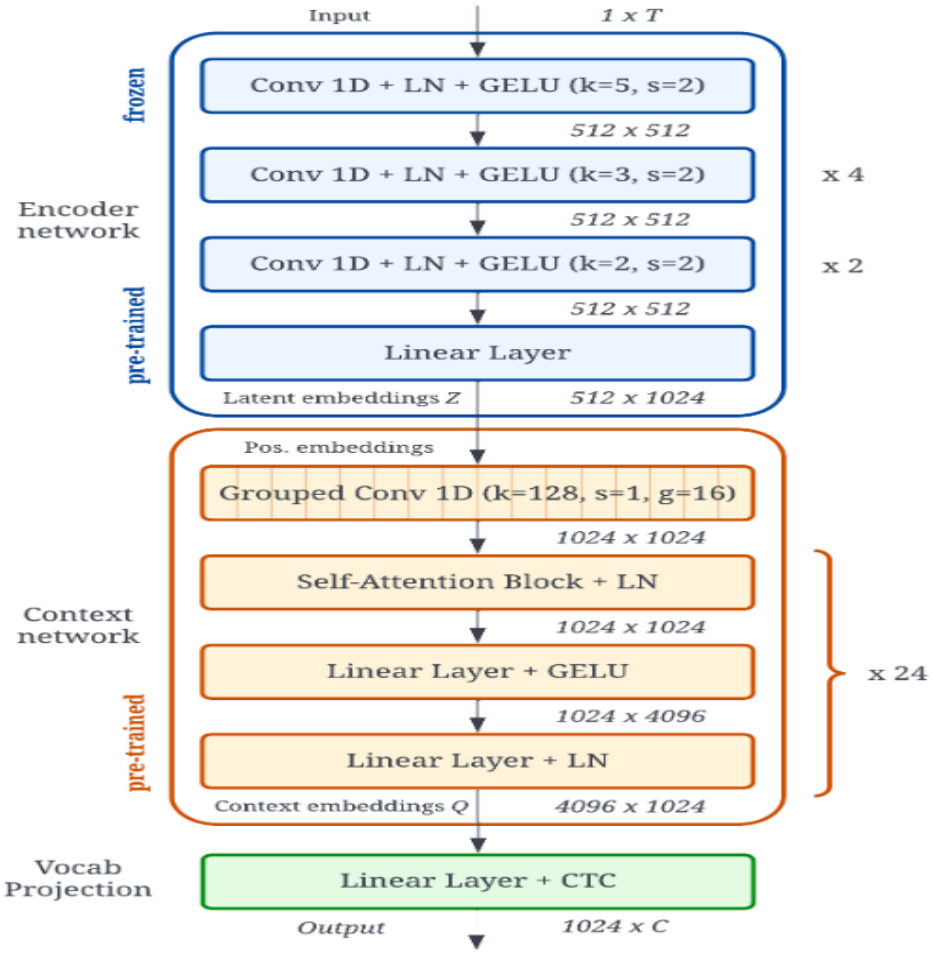

Figure 32. The proposed ASR architecture (Thienpondt and Demuynck, 2022).

The model applies a bi-gram in-domain LM to decode the utterances as described by Shahnawazuddin et al (2020). The evaluation of the proposed model was done through PF-STAR British English Children's Speech corpus. And referring to the experimental results, in the case where the LM is not applied, a WER of 6.57% is obtained. And in the case when the LM is applied, the WER reaches 4.86%.

Bao et al (2019) have proposed various methods of acoustic modelling and data augmentation to improve the accuracy of a deep learning ASR framework for a low-resource language. They presented five acoustic models and three data augmentation techniques. The first acoustic model is built based on the GMM/HMM framework implemented in the Kaldi toolkit. It integrates the HMM/GMM baseline

architecture with a triphone model, and a trigram KenLM language model (Kenneth and Kenlm, 2011). The second and third acoustic models are built over the DNN-RNN framework of DeepSpeech (Awni et al., 2014). They consist of 5 RNN layers integrated with LSTM cells. The first three layers and the fifth layer are fully connected, while the fourth is a BiRNN layer. These approaches are trained based on the CTC loss model (Alex et al., 2006). The fourth and fifth acoustic model consists of a 1D gated convolutional neural network as described by Liptchinsky (Vitaliu et al., 2017). The proposed model is called mini-GCNN and is derived from the architecture described by Liptchinsky et al (2020).

There are applied three data augmentation models: the first consists of adding data to the training corpus to adjust the distortion of the input speech and reduce the background noise as described by (Robbie et al., 2018); the second and third are based on voice conversion.

Very important in this framework is the training process, where is implemented a multistage transfer learning strategy related to data augmentation goes through three stages as shown in Figure 33.

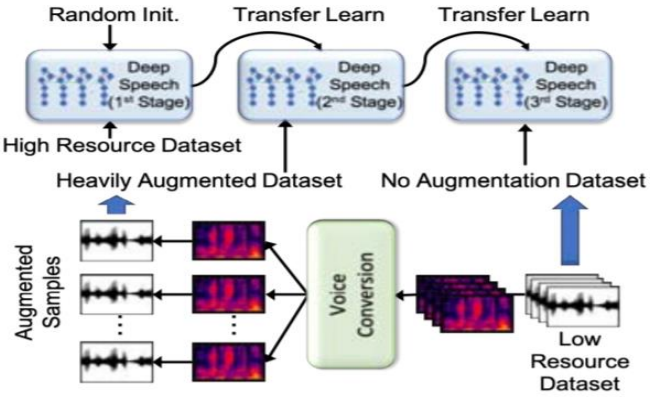

Figure 33. Multistage training of DeepSpeech using transfer-learning (Bao et al, 2019).

During the first phase, the model (DeepSpeech or mini-GCNN) is trained with LibriSpeech corpus randomly. During the second phase, the model is trained with the best weights generated during the first phase. And during the third phase, the model is trained with the weights generated during the second phase and outputs the final representations.

The application of data augmentation techniques significantly increases the amount of data compared to the original data. And referring to the experimental results the application of these techniques reduces the WER by 15%.

The consulted papers in identifying potential advanced ASR systems for low resource language have been included in Table 3. As can be shown in Table 3, the architectures that we have analyzed are comprehensive, including hybrid systems, end-to-end systems as well as their combination with Transformers. All selected models are from the last five years. We noticed that in the last two years speech recognition technology for low-resource languages has been mainly focused on end-to-end models and Transformers.

Table 3: Analysis of advanced ASR systems for low resource language.

Authors	Hybrid	End-to-End	Deep Learning	TDNN	GMM-HMM	DNN-HMM	BiGRU	Augment	Transfer Learning	CNN-LSTM	Transformers	wav2vec	Conformer
Vegesna et al (2017)	√		√		√	√							
Fathima et al (2018)	√		√	√		√							
Zhou et al (2018)		√	√								√		
Baevski et al (2019)			√								√	√	
Chen and Yand (2020)	√	√	√	√	√	√							
Wang et al (2020)		√	√							√			
Anoop et al (2021)		√	√				√	√		√			
Meng et al (2021)		√	√								√		
Xue et al (2021)		√	√								√		
Baevski et al (2021)		√	√								√	√	
Guo et al (2021)		√	√					√			√		√
Thienpondt and Demuynck (2022)			√					√	√		√	√	

Referring to the analysis that we have done for each paper, each architecture has its advantages and disadvantages. But the aim of our study at this phase is focused on the overall performance of the system, referring to the standard performance measurement parameter word error rate (WER). In Figure 34, we graphically presented the value of WER for each architecture.

First, we emphasize that each architecture is trained and evaluated with a specific corpus related to the language for which it was designed. Several architectures have been trained and tested with several corpora and we have presented in the graph the architecture with the best performance. Since corpus size, corpus accuracy, speaker attributes and other corpus parameters are not the same, all these factors jointly directly affect the accuracy of the model.

We also emphasize that each language has its own morphological, syntactic and linguistic characteristics, making the comparison of architectures more complex. Regardless of all these factors the results presented in Figure 34 clearly show which architectures have the best performance.

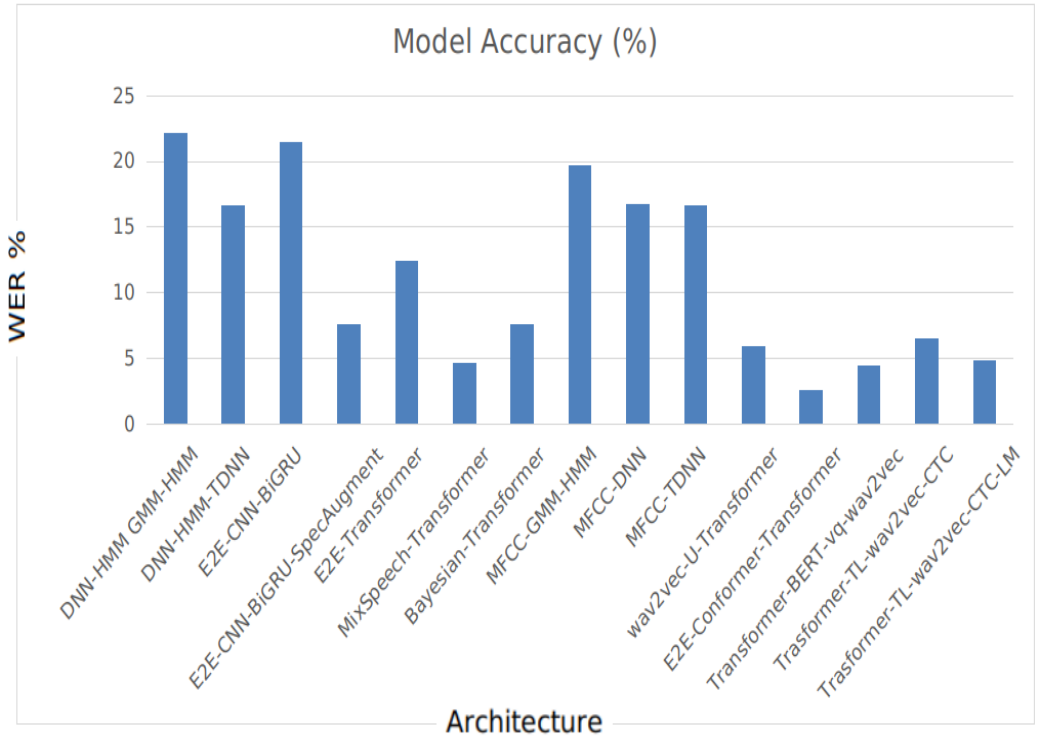

Figure 34. The WER values for all end-to-end architectures were analyzed.

Referring to the results presented in Figure 34, we conclude that the architectures that perform better for low-resources languages are mainly based on the Transformers models. We noticed that the application of data augmentation and transfer learning techniques, and in particular wav2vec approaches, significantly increase the accuracy of the model. Also, end-to-end techniques based mainly on RNN and their variants such as CNN, BiRNN, etc., present very promising results, especially when applying different data augmentation techniques. While hybrid systems present relatively good results for low-resource languages.

2.6. Automatic Speech Recognition (ASR) Systems based on Transformers.

Transformers are a powerful deep learning model which consists of two parts: an encoder which works on the input sequence and a decoder which operates on the target output sequence (Vaswani et al., 2017). The Transformers work through sequence-to-sequence learning where the Transformers takes a sequence of tokens and predicts the next word in the output sequence. It does this through iterating encoder layers, so the encoder generates encoding that defines which parts of the input sequence are relevant to each other. Then passes these encodings to the next encoder layer. The decoder takes all of these encodings and uses their derived context to generate the output sequence. Transformers are considered a form of semi-supervised learning, which means that they are pre-trained in an unsupervised manner with a large, unlabeled data set, and they are fine-tuned through supervised training to get them to perform better. What makes Transformers a little bit different is that they do not necessarily process data in order. Transformers use something called an attention mechanism which provides context around items in the input sequence. So rather than start to run the first word of the sentence, the Transformers attempt to identify the context that brings meaning to each word of the sequence. And it is this attention mechanism that gives transformers a huge leg up over algorithms like RNN that must run in sequence. Transformers run multiple sequences in parallel and this vastly speeds up training times. Figure 35 presents the baseline architecture of the Transformers with all its components.

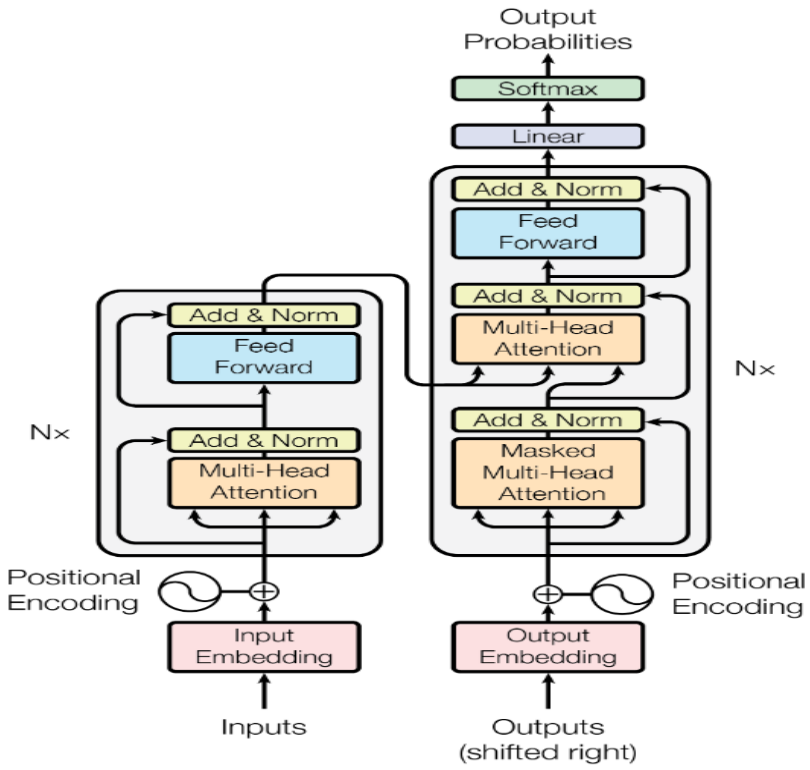

Figure 35. The Transformers model architecture (Vaswani et al. 2017).

The two main blocks of the Transformers-based architecture are the encoder and the decoder. The encoder is composed of some identical layers which are built of the same type of blocks. Each layer includes a multi-head self-attention mechanism (MHA) and a position-wise fully connected FFN. Around these mechanisms is employed a residual connection followed by layer normalization. The output of each mechanism is Layer Norm(x + Sub layer(x)), where Sub layer(x) is the function implemented by the mechanism itself.

In the same form as the encoder, the decoder also consists of several identical layers. In addition, it placed a third sub-layer, to perform MHA at the output of the encoder. Even in the decoder, a residual connection is followed by layer normalization. One of the essential elements of the baseline architecture of Transformers is the multi-head attention mechanism. It applies self-attention multiple times in parallel using different weight matrices. As can be shown in Figure 36, in the centre of architecture is the scaled dot-product attention which is repeated h times. So, the linear corresponds to megaesophagus to the matrix multiplication between the weight matrices and the inputs. The concat module repeats multiple times the scalar product attention each time with different weights matrices

and then concatenates the results. Next is the linear module. It represents a linear matrix which provides more parameters for learning, instance, and the inputs.

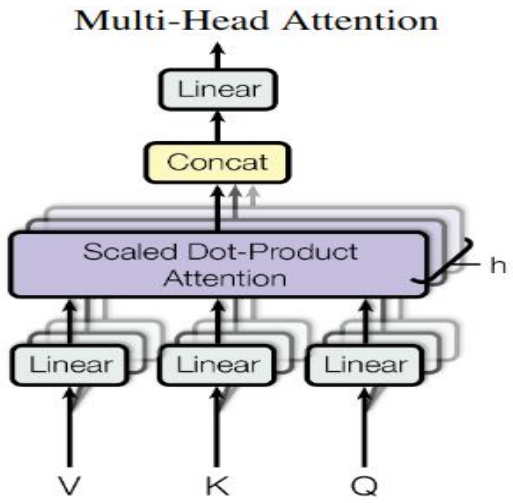

Figure 36. Multi-head attention (Vaswani et al. 2017).

Multi-head attention enables the model to process data from different representations. Matrix of outputs will be given as follows:

$$\mathrm{MultiHead}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{Concat}(\mathrm{head}_1 \dots \dots \mathrm{head}_\mathrm{h})\mathrm{W}^\mathrm{o}$$

Where, $\mathrm{head}_\mathrm{i} = \mathrm{Attention}(\mathrm{QW}_\mathrm{i}^\mathrm{Q}, \mathrm{KW}_\mathrm{i}^\mathrm{K}, \mathrm{VW}_\mathrm{i}^\mathrm{V})$

And the projections are parameter matrices $\mathrm{W}_\mathrm{i}^\mathrm{Q} \in \mathrm{R}^{\mathrm{d_{model}\ xd_k}}$, $\mathrm{W}_\mathrm{i}^\mathrm{K} \in \mathrm{R}^{\mathrm{d_{model}\ xd_k}}$, $\mathrm{W}_\mathrm{i}^\mathrm{V} \in \mathrm{R}^{\mathrm{d_{model}\ xd_v}}$, $\mathrm{W}^\mathrm{O} \in \mathrm{R}^{\mathrm{hd_v\ xd_{model}}}$

In this session, we will analyze the most cited and popular Transformers-based ASR systems during the five last years. For each selected architecture, we will make a detailed analysis of it, describing all the components with which it is built. In addition, we will analyze the performance of each architecture by referring to the international standard word error rate (WER) parameter. At the end of this session, we aim to define the best Transformers-based architecture.

Dong et al (2018) have proposed a Speech-Transformers ASR model based on the seq2seq approach. It relies on attention mechanisms that transform input speech data to the corresponding character

sequences. In addition, they propose a 2D-Attention mechanism that replaces the time-frequency recurrence with both the temporal and spectral dependencies captured by attention.

The Speech-Transformers rely on the encoder-decoder baseline architecture, and it is composed of two main modules: 1) the multi-head attention; and 2) the position-wise feed-forward network.

The multi-head attention aims to leverage various attending representations jointly. It applies a scaled dot-product attention mechanism as described by Vaswani et al (2017) that connects different input positions to compute representations of them. Then, these representations are fed into a linear projection to obtain the final dimensional outputs.

The position-wise FFN is composed of two linear transformations and a ReLU between them. Figure 37 shows the full Speech-Transformers architecture.

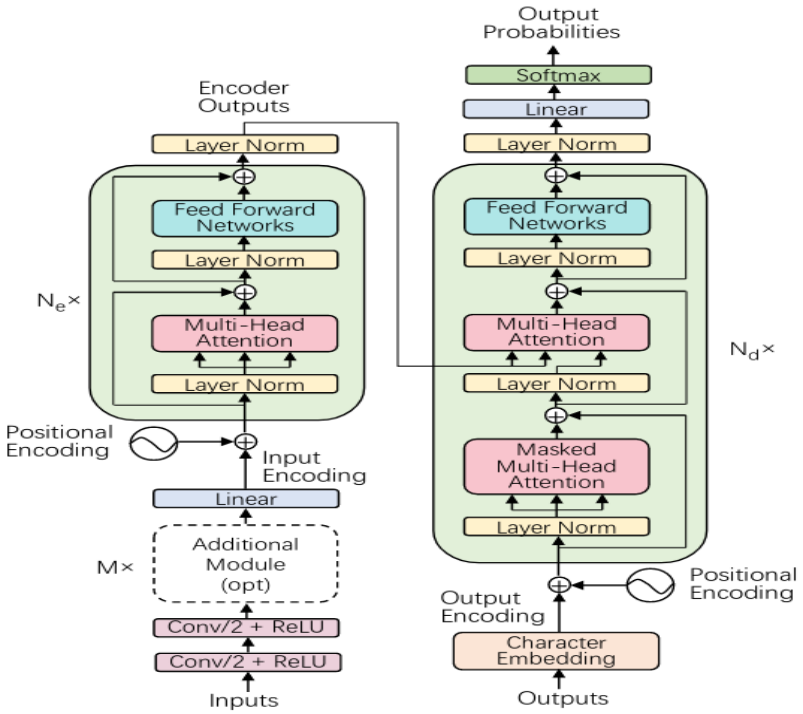

Figure 37. The Speech-Transformers architecture (Dong et al., 2018).

The encoder is composed of two 3×3 CNN layers with stride 2 for both time domain and frequency domain dimensions to avoid overflow and produce the hidden representation length according to character length, as can be shown on the left of Figure 37. Next, according to the parameters of the model, M additional modules can be applied to extract more representations. Then, a linear

transformation is applied to obtain vector input encoding features. These features feed the input encoding to apply relative position rules. The final output is obtained by processing the sum of input encoding and positional encoding by several encoder blocks. Each encoder block is composed of a multi-head attention sub-block and the position-wise feed-forward network sub-block. In each of these sub-blocks are applied a layer normalization and residual connection to improve the training process. Each decoder-block is composed of three sub-blocks: The first sub-block is masked multi-head attention, which ensures the prediction position depends on the results with the same position or with a smaller position. The second is multi-head attention and the third is another position-wise feed-forward network. Even the decoder in each of its sub-blocks applies a layer normalization and residual connection. The output of the decoder passes through a linear projection and a softmax function to produce the final distribution of probabilities.

Another innovation that this architecture brings is the application of the 2-D attention mechanism. This mechanism connects positions on both the time and frequency axis to build the temporal dependencies. First, it performs 3 convolutional networks to extract the representations of queries, keys and values independently. Then, it applies the scaled dot product attention mechanism to capture temporal and spectral dependencies. The last, the outputs of this mechanism are collected and fed to another convolution network, which produces the final outputs. Training and evaluation of the model are done through the WSJ corpus, which yields a WER of 10.9%.

Hrinchukl et al (2019), have proposed a Transformers-based encoder-decoder architecture, which is based on Jasper (Li et al. 2019), a deep convolutional E2E model as shown in Figure 38.

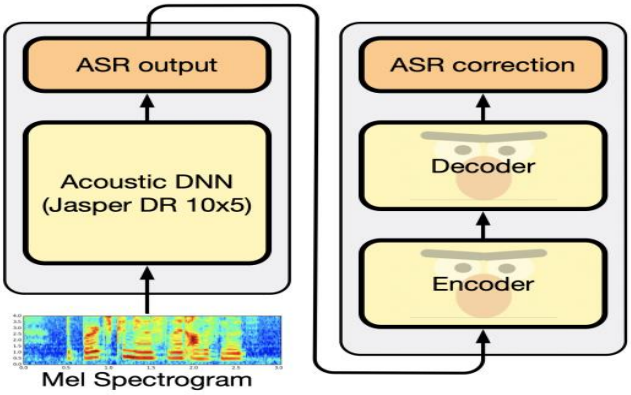

Figure 38. ASR correction Transformers-based encoder-decoder architecture (Oleksii et al. 2019).

Jasper folds the AM and PM into a single CNN. It takes as input Mel-filter bank features and generates a probability distribution of characters. The model is trained with CTC loss function, while the baseline Jasper model is trained with the Novograd [Ginsburg et al. 2019] optimizer implemented in Pytorch. As language models, authors have considered 6-gram KenLM [Heafield et al. 2011] and Transformers- XL [Dai et al. 2019], which complement each other. The basic components on which the proposed architecture is built are derived by Transformers encoder Decoder baseline architecture [Vaswani et al, 2017], where the fully-connected inner-layer dimensionality is set to 4H. H represents the hidden sizes. To initialize the model weights, are used random initialization and using the weights of pre-trained BERT [Devlin et al. 2019]. The BERT parameters are used for encoder initialization, while to initialize the decoder, the parameters of the corresponding self-attention block are used. The training and evaluation of the proposed model are done through the LibriSpeech corpus. Referring to the experimental results, the proposed model when it applies the Transformers-XL language model achieves the best performance by achieving a WER of 2.95% in test-clean and 8.79% in test-others.

Moriya et al (2020) have proposed a CTC-Transformers -based ASR system, which applies a novel training technique based sequence-to-sequence (S2S) approach. The S2S approach uses the attention mechanism to capture connections between input and output sequences, by informing which time frames have to be attended for predicting the output labels. The innovation of this model is the application of a self-distillation mechanism to improve the performance of CTC- Transformers baseline architecture (Karita et al., 2019). This mechanism creates an attention matrix, which is composed of Transformers output and the attention weights of each head. The number of attention matrices created will be the same as the number of attention heads. In addition, an additional loss called SD is applied that represents the CE loss between the attention matrix and the output of the encoder (Moriya et al., 2018). The SD loss helps to train the shared encoder. Figure 39 shows the schematic diagram of the proposed model, and the proposed objective is shown with dashed red lines.

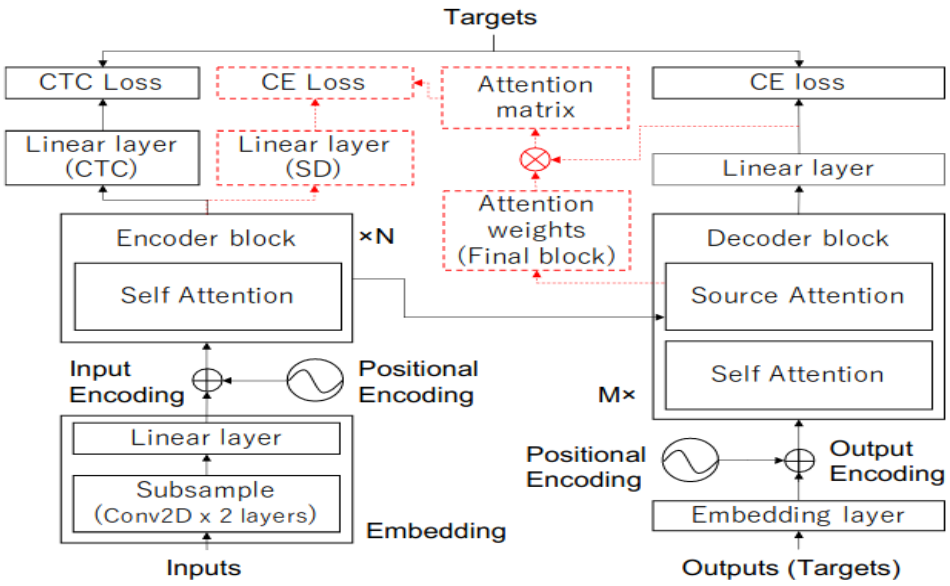

Figure 39. The schematic diagram of our CTC-Transformers model (Moriya et al., 2020).

For training and decoding, the Transformers and CTC rules are applied (Karita et al., 2019). During the training process, the models are trained in a way to minimize the losses for both Transformers and CTC losses jointly as can be shown with solid boxes in Figure 39. While during the decoding process, the probabilities of the Transformers have been combined with the CTC and language model probabilities as described by Hori et al (2017).

The evaluation of the proposed model is done through five different corpora, but we have only focused on the analysis of the results obtained from LibriSpeech. And specifically, the model yields a WER of 2.4% in test-clean and 5.6% in test-others.

Baevski et al (2020) have proposed a wav2vec framework for ASR based on self-supervised learning. This framework encodes speech signals through a convolutional network and masks them. The masked features fed Transformers which built the contextualized features. Figure 40 shows the structure of this framework.

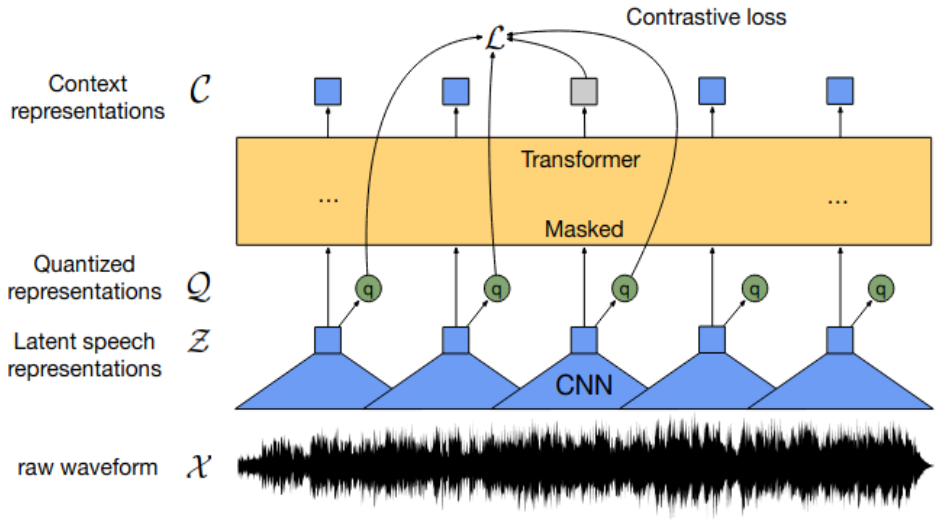

Figure 40. The proposed architecture (Baevski et al., 2020).

This model consists of a multi-layer convolutional encoder which is fed by speech audio waves and produces latent speech representations. The encoder is composed of multiple blocks which contain a temporal convolution, a layer normalization as well as a GELU activation function (Jiang et al., 2019). Then, they are fed to Transformers, which builds representations by capturing information from the entire sequence as described by Baevski et al (2019). The output of the feature encoder passes into the quantization module to produce the context representations which are built over all continuous speech representations. The quantization module applies a Gumbel softmax which selects discrete codebook entries in a differentiable way (Jang et al., 2016). The model first is pre-trained by masked latent features which represent the quantized latent speech features. Next, the pre-trained models are adapted for SR by applying an initialized linear projection into the context network (Baevski et al., 2019). In addition, the model is improved by applying the CTC loss function (Graves et al., 2006) as well as a Spec Augment technique (Park et al., 2019). All models are implemented in fairseq (Ott et al., 2019), and use the Adam optimizer technique (Kingma et al., 2015). Also, they have implemented two language models: the 4 - gram LM and the Transformers LM. The training and evaluation of the model are done through the LibriSpeech corpus. Referring to the experimental results, the best WER results were obtained when Transformers-LM is applied, yielding a WER of 1.8% in test-clean and a WER of 3.3% in test-others.

Baevski et al (2020) have proposed a model, which is based on BERT baseline architecture, which learns directly by continuous speech data. In addition, they proposed an improved version of BERT architecture which applies a CTC approach.

The proposed model is built over the architecture proposed by Baevski et al (2020) where speech data are quantized using a contrastive loss. It uses the vq-wav2vec quantization. Which applies a Gumbel-softmax technique as described by Baevski et al (2020). The BERT architecture that is applied derives from (Delvin et al., 2019), which is trained only with the masked language modelling feature on each sequence. In addition, the positional embedding in the BERT architecture is replaced with a single group convolutional layer, which is applied directly to the embedding (Mohamed et al., 2019).

Another innovation in this architecture is that the inputs of the model are dense wav2vec features, MFCC or FBANK features, where some of them are replaced with a mask embedding to feed the Transformers encoder. The model is optimized by applying the InfoNCE loss as described by Oord et al (2018). In addition, a linear projection is attached to the representation features to further improve the model. To improve the accuracy and reduce training parameters a Spec Augment technique is applied (Park et al., 2019). Also, a channel masking is applied, which determines the channel index and its width. In addition, a dropout in each layer of the Transformers is applied to improve regularization and stability. Both proposed models based on quantized inputs training and based on continuous inputs training are implemented in the fairseq toolkit (Ott et al., 2019), and are trained and tested using the LibriSpeech corpus. Referring to the experimental results, the discrete BERT model when applying vq-wav2vec has the best performance, yielding a WER of 4.5% in test-clean and a WER of 12.1 in test-others.

Dalmia et al (2021) proposed an end-to-end ASR system for code-switched speech recognition that uses a Transformers-transducer model architecture. This architecture is built on the development of the vanilla model making three main improvements 1) creation of two auxiliary loss functions that handle the low-resource scenario of code-switching; 2) proposal of a mask-based training technique that incorporates language ID information to enhance label encoder training for intra-sentential code-switching, and 3) proposal of a multilabel/multi-audio encoder structure to make use of monolingual speech corpora to code-switch. Figure 41 shows the proposed architecture.

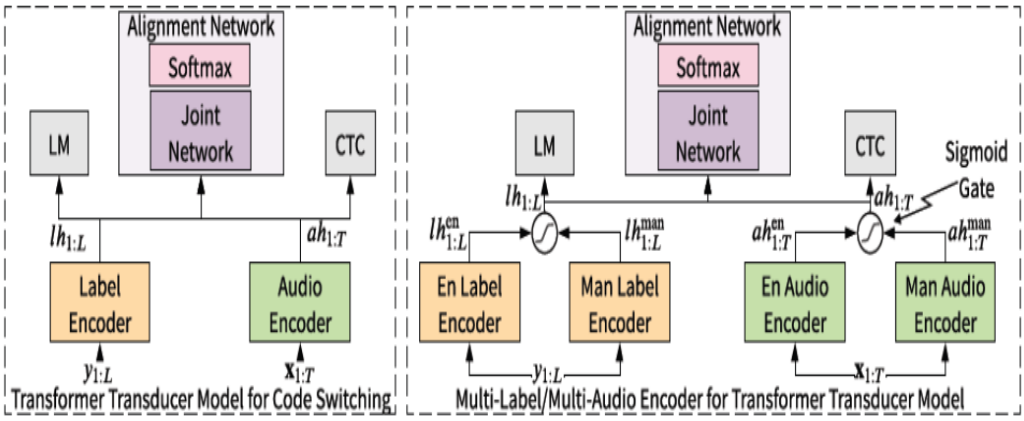

Figure 41. Proposed Transformers-transducer model for code-switching (Dalmia et al, 2021).

This model is based on a neural transducer that makes the output reliant on all prior labels. This model learns explicit input-output alignments, making them resilient to extended utterances. Attention is focused on extending the Transformers-transducer (T-T) concept to code-switched ASR in particular. RNNs are replaced with non-recurrent multi-head self-attention Transformers encoders in the T-T model. The three main components with which the transducer is built are 1) label encoder; 2) audio encoder, and 3) alignment network. The label encoder aims to encode the output sequence. The audio encoder aims to encode the input audio representations. While the alignment network aims to create the output lattice based on the encoded output sequence and encoded input audio representations. To train the model a vanilla T-T model is used. As data augmentation technique authors have proposed three-way speed perturbation [Povey et al., 2011] and Spec Augment [Park et al., 2019]. The audio encoder's job is to learn audio representations at the frame level. Similarly, the label encoder's job is to encode the previous context, which is utilized to forecast the following word during transducer loss alignment. The final output is the function which uses the two auxiliary tasks as follows:

$$\mathrm{F_{obj}} = \mathrm{F_{Transducer}}(\mathrm{x},\mathrm{y}) + \lambda_{\mathrm{CTC}}\mathrm{F_{CTC}}(\mathrm{x},\mathrm{y}) + \lambda_{\mathrm{LM}}\mathrm{F_{LM}}(\mathrm{y})$$

Where λ_{CTC} and λ_{LM} function as tuneable weights for the auxiliary tasks.

The model is trained with the SEAME, a public Mandarin-English code-switching corpus, and it achieves great results.

Chen et al (2020) proposed two non-autoregressive Transformers architectures for automatic speech recognition (ASR). Audio-Factorized Masked Language Model (A-FMLM) and Audio-Conditional Masked Language Model (A-CMLM). In both models, the decoder is fed by masked token representations. To predict these representations, both unmasked token representations and input row audio waves must be taken into consideration. Both of these models can apply different decoding techniques. The language model's training approach is comparable to BERT [Delvin et al., 2019]. The network is learned to anticipate original tokens by replacing some random tokens with a particular mask token. This approach differs from BERT in that the system would make predictions based on the voice input. The decoder uses each subset to predict the input features. It begins with an empty set and works our way through the entire process. Because the subset picked is so adaptable, any decoding order is acceptable.

Evaluation of the models is done on Aishell (Bu et al., 2017) and Spontaneous Japanese (CSJ) (Maekawa, 2003) corpora. The encoder is composed of 12 Transformers blocks, where each block consists of some convolutional layers that aim to reduce the sampling rate. Six Transformers blocks make up the decoder. Four heads are employed to pay attention to all Transformers blocks. The warm-up is used for early development stages after the network has been trained. All experiments employ a beam search and language model, with all configurations adhering to an auto-regressive baseline.

Referring to experiment results, all of the decoding approaches produce results that are quite near to those of state-of-the-art autoregressive models. The performance of the A-FMLM is close to the autoregressive baseline architecture, by achieving a CER of 5.4%. This is because, unlike the autoregressive model, the non-autoregressive systems only conduct decoder calculations a certain number of times.

Mohamed et al (2020) have proposed a convolutional context Transformers-based model. It replaces the sinusoidal positional embedding with convolutional learned input representations. These representations reorganize the Transformers layers according to the intended order and ensure their local learning.

The innovation of this model consists of the organization of learning in two contexts: 1) local learning in convolutional layers which are located below Transformers layers; and 2) global learning in Transformers layers. The structure of the proposed Transformers layers is based on the model described by Vaswani

et al (2017). The core of them is considered the multi-head self-attention mechanism, which has a large range of actions enabling global sequential learning over the input speech. The encoder is composed of 2-D convolutional blocks, where each of them consists of several layers and a max pooling layer. In addition, it applies a ReLU as well as a layer normalization after each convolutional layer. The decoder has the same structure as the encoder, but instead of 2-D convolutional blocks, it uses 1-D convolutional blocks over the embedding of previously predicted words. Figure 42 shows the full proposed architecture. On the left is presented the component of the Transformers block, and on the right is presented the full end-to-end ASR architecture.

The decoder applies a multi-head attention layer to collect encoder context representations, which will significantly improve the recognition performance of the model. The 1-D convolution block in the decoder takes into account only the final predictions that correspond to the current time step. The decoder self-attention operates over current and previous time steps, while the encoder self-attention operates over the entire input utterance. The Transformers is composed of two 2-D convolutional blocks, where each block has two convolution layers, a kernel Size = 3, and a max-pooling kernel = 2. The first block represents 64 feature representations maps while the represents 128. For regularization, a dropout rate of 0.15 across all blocks is applied, and for model optimization, is applied the AdaDelta algorithm (Zeiler, 2012).

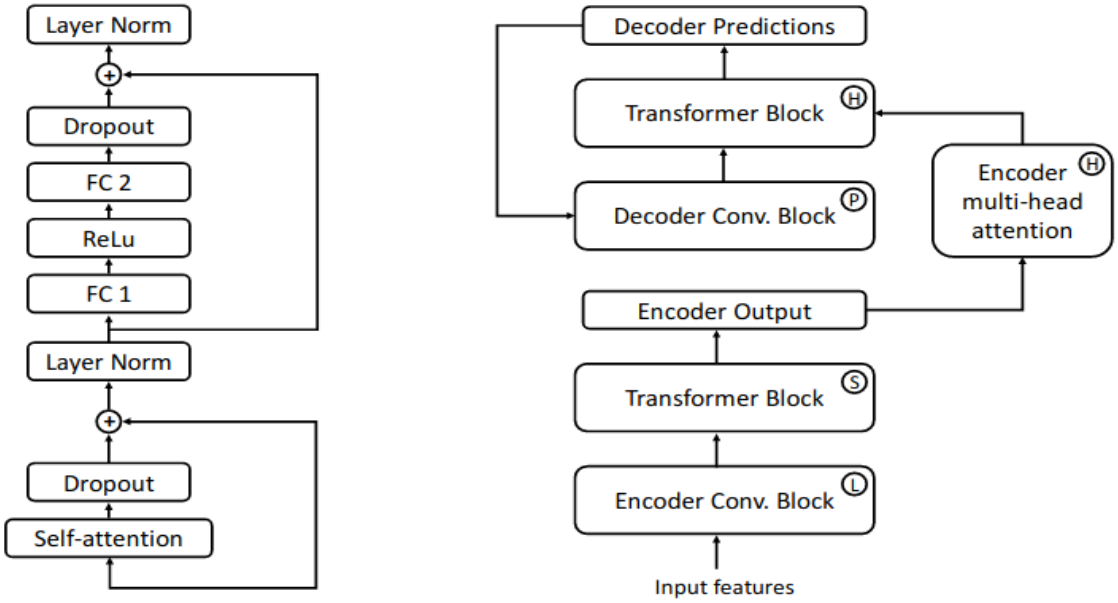

Figure 42. Full end-to-end ASR architecture (Mohamed et al., 2020).

The model is trained and evaluated with LibriSpeech corpus and achieves great results. It yields a WER of 4.5% in test-clean and 12.9 % in test-others.

Another architecture in this domain is presented by Fan et al (2021). They have proposed a CTC alignment-based single-step non-autoregressive Transformers (CASS-NAT) model for speech recognition. This model is based on the hybrid CTC-Attention architecture with Transformers structure as is shown in Figure 43. The queries, keys, values and mass matrix, represent the Bi-Mask in Encoder, the Trigger Mask in Token Acoustic Extractor, and the Bi-Mask in Decoder. On the top of the encoder, is located a CTC loss function and all the output sequences from CTC are considered as alignment. The attention mask is applied to maintain the attention range within the allowed limit for each output. The token acoustic extractor module extracts each token in parallel with the CTC alignment features. The CTC alignment features represent both the time-space of the acoustic features and the number of tokens which will feed the decoder. To extract the token features both the trigger mask and the sinusoidal positional encoding are applied. Since the decoder does not need to create recurrence, a bidirectional mask is used, which creates the relations between token-level acoustic embedding.

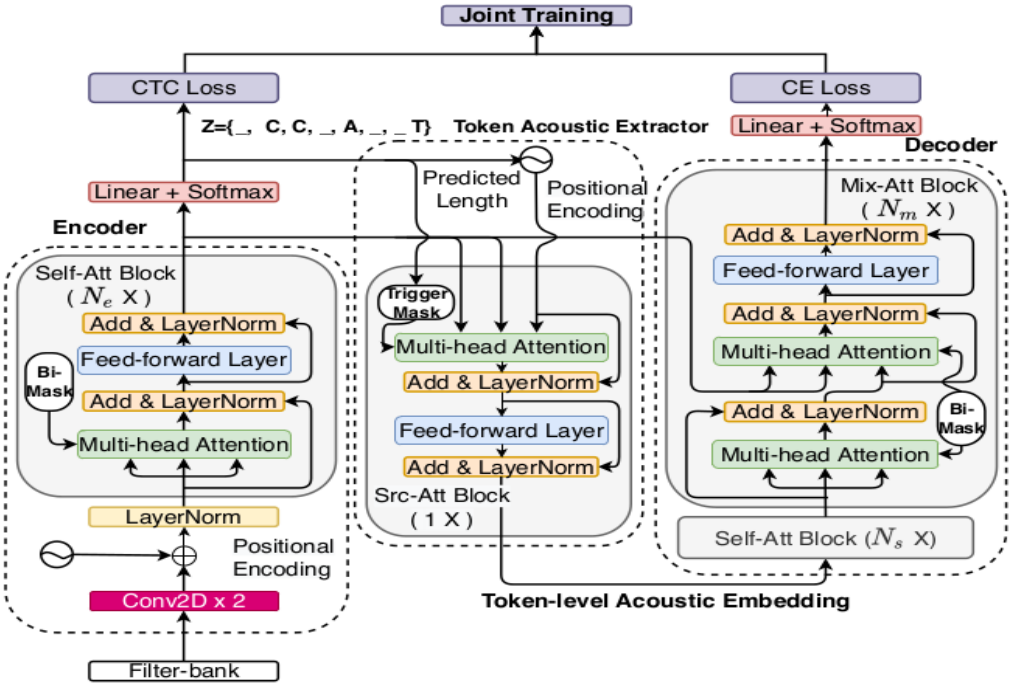

Figure 43. The CASS-NAT architecture (Ruchao et al 2021).

The training and evaluation of the model are done on the LibriSpeech corpus. Referring to the experimental results, this architecture yields a WER of 3.8% in the test clean and 9.1% in the test-others. While in the case when a language model based on Transformers is applied, the WER goes to 3.3% in test-clean and 8.1% in test-others.

Fan et al (2021), have proposed several techniques to improve the accuracy of the CASS-NAT baseline architecture. First, they proposed to add convolution augmented self-attention blocks to both the encoder and decoder modules, which aims to model the local dependencies of the input sequence in the encoder as described by Yang et al (2019). Specifically, the feed-forward layer is divided into two sublayers where one is placed at the beginning of the block and the other at the end. In addition, a convolution layer is placed after the self-attention layer to be used as a normalization layer and has the same aim as batch normalization. It applies a BiMask for both mix-attend decoder and self-attend decoder directions.

Second, they propose to expand the acoustic boundary (trigger mask) for each token to stabilize the CTC alignments. This provides compensation for the inaccuracy of token-level acoustic embedding extraction as well as the contextual frames for each token. Then, the acoustic boundary will be further expanded with the extraction of acoustic embedding.

And the third improvement deals with the application of a loss function repeatedly to increase the variables of low layers. Thus avoiding the problems that come as a result of gradient vanishing. For this purpose the CTC loss function (Lee and Watanabe, 2021) is integrated into the iterated CE loss functions (Wang et al., 2020) by providing that the parameters in different layers can be updated at the same scale. The full proposed architecture is shown in Figure 44.

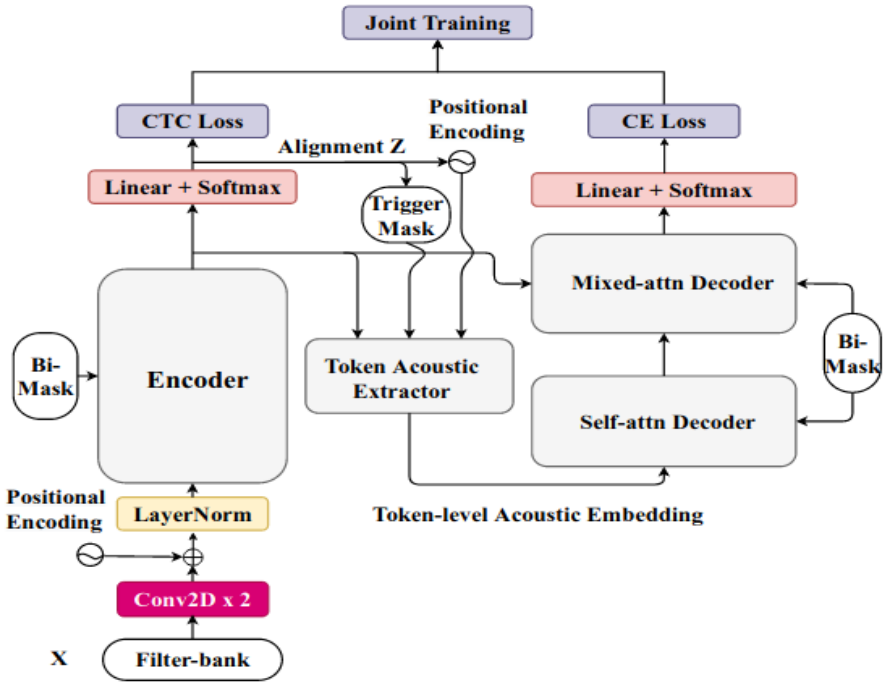

Figure 44. The CASS-NAT architecture (Fan et al., 2021).

The evaluation of the proposed model was done through the LibriSpeech corpus, without applying any language models. Referring to the experimental results, this model yields a WER of 4.9% in test-clean and 5.4% in test-others.

Shi et al. (2021), have proposed a streaming Transformers transducer model based on processing the middle block and look-ahead context independently using non-causal convolution [Yeh et al., 2020]. This approach uses look ahead context in convolution to enhance the streaming Transformers converter for voice recognition while maintaining the same training and decoding performance.

The proposed architecture is based on an advanced Enformer model (Shi et al., 2021). Emformer uses parallel block computing to split an input sequence into numerous non-overlapping blocks to facilitate streaming speech recognition: $C_i^n, \ldots\ldots, C_{i-1}^n$, where i represents the current block's index and n represents the layer's index. Enformer is based on the principle of dividing an utterance into multiple sequences. The present sequence, as well as its surrounding left and look ahead contexts, are computed by self-attention. Figure 45 shows the Emformer and advanced Emformer architecture.

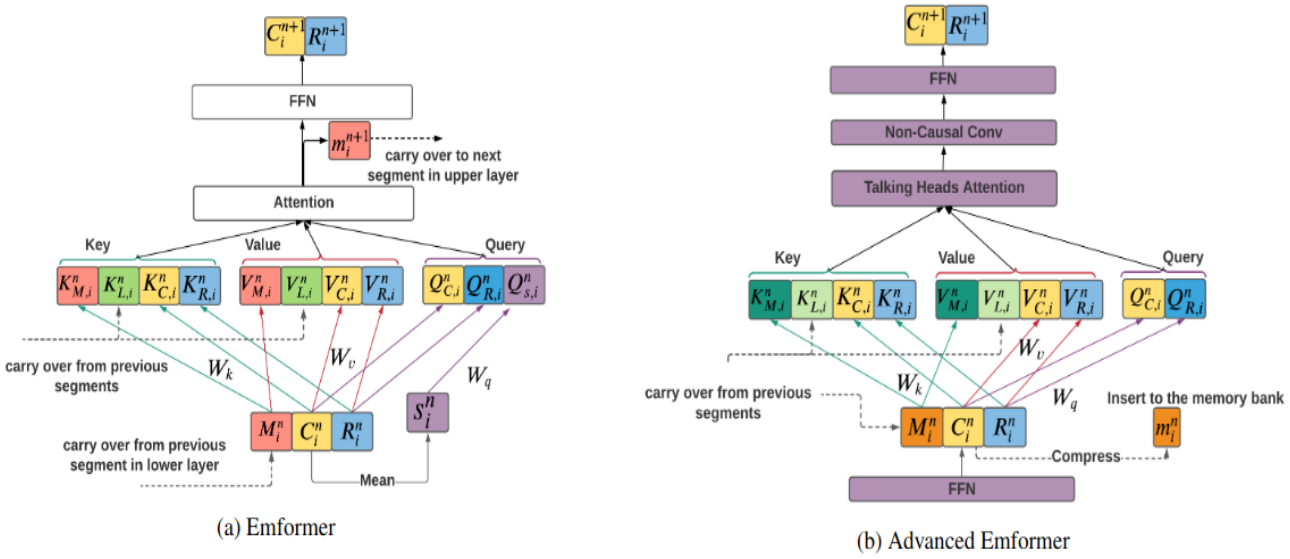

Figure 45. Advanced Emformer architecture with non-causal convolution (Shi et al., 2021).

The encoder is composed of 21 Emformer layers. For self-attention, each layer has four heads and an FFN-block dimension of 1280. For depthwise convolution operations, a kernel size of 7 is employed. The superframe is mapped to a 256-dim vector when using the 32M parameter model. The superframe is mapped onto a 384-dim vector using the 73M parameter model. Each layer has an 8-head self-attention block and a 1456-dim FFN block. Each layer has a 4-head self-attention block and a 1024-dim FFN block. The evaluation of the model was done using the assi, call and dict corpora. The assi and call are composed of 13 600 utterances, while the dict is composed of 8 hours of speech data. Referring to the experimental results, the best result was achieved when the model was trained with the ass corpus and it yields a WER of 4.66%.

Wang et al (2021) have proposed self-supervised learning (SSL) ASR, which applies intermediate layer supervision (ILS) so that the model focuses on content information as much as possible. In this way, the losses are calculated on the top layer as well as on the intermediate layers, making the lower layers have to learn more about content information to optimize the intermediate SSL loss. The proposed model is based on the HuBERT baseline architecture (Delvin et al., 2019) as is shown in Figure 46. It is composed of a convolutional encoder and a Transformers encoder. The encoder consists of 7 convolution blocks, a normalization layer as well as a GELU layer. While the Transformers encoder is composed of a

convolution-based relative position embedding layer. In addition, a bucket relative position embedding is added to encode the position information (Raffel et al., 2020). The ILS provides lower layers to learn content information. During the pre-training of the model, some layers are selected as supervised layers, which calculate the masked prediction loss of the output hidden states. After pre-training, all the prediction layers are extracted and the model is improved with the CTC loss function, which is applied to the top Transformers layer.

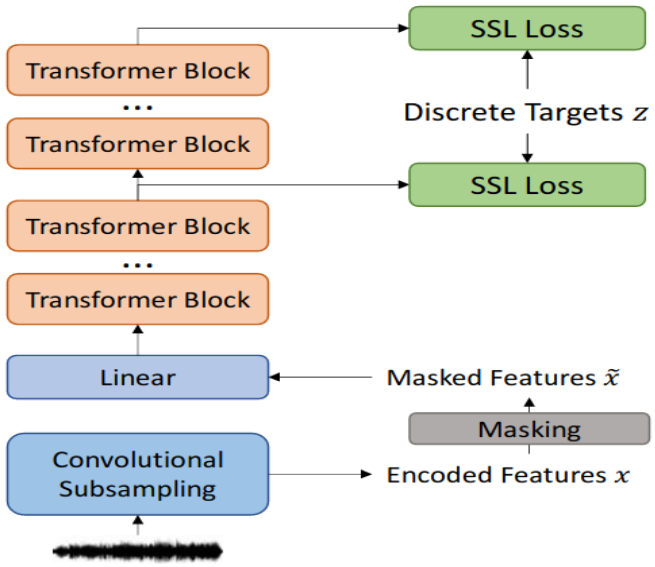

Figure 46. Proposed architecture (Wang et al., 2021).

The pre-training process and all hyper-parameters are configured as described by (Hsu et al., 2021). The model is optimized according to the AdamW optimizer technique. The training and evaluation of the model are done using the LibriSpeech corpus. Referring to the experimental results when the model applies the 4-gram LM yields a WER of 3% in test-clean and 6.9% in test-others. When LM is not applied in test-clean WER reaches 4.7% and in test-others 10.1%.

Deng et al (2022), have proposed a non-autoregressive (NAR) CTC/attention model which applies both pre-trained acoustic and language models, the BERT and wav2vec2.0. One of the innovations of this architecture is the application of a MCM mechanism which aims to connect acoustic and linguistic features and overcome the barrier between speech and text representation. Figure 47 shows all components of the proposed architecture.

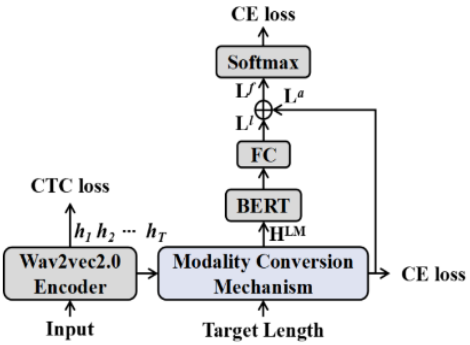

Figure 47. The proposed NAR CTC/attention architecture (Deng et al., 2022).

This architecture is composed of three main components: 1) the MCM mechanism; 2) a pre-trained wav2vec2.0 encoder, and 3) a pre-trained BERT. The MCM mechanism is composed of two main modules:

The first module applies a CTC branch integrated with a greedy algorithm to predict the targeted sequence. Over the target sequence a position encoding is applied, to match its dimensions with acoustic features. Then, the generated features are combined with acoustic features and fed a MHA block, which produces an alignment of characters.

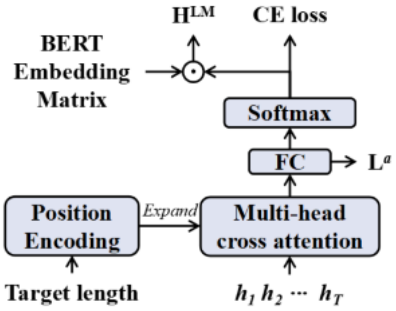

Figure 48. The components of the modality conversion mechanism (MCM) (Deng et al., 2022).

The alignment converts the acoustic representations into a length embedding as described by Yu and Chen (2021). In this case, it is not necessary to increase the training iterations.

The second module calculates the sum of the token embedding representations at BERT architecture. Then the linguistic representation is extracted which is fed into the BERT encoder. As can be shown in Figure 48, to produce correct predictions a cross-entropy (CE) feature is applied followed by a softmax function. This module represents powerful text processing capabilities provided by BERT architecture.

The wav2vec2.0 encoder is composed of a CNN and a Transformers network. It is used to extract an acoustic representation from the raw audio wave. At the top of the BERT, the output is applied a FC layer aims to obtain the linguistic representations.

Evaluation of the model is done through the Mandarin AISHELL [Bu et al., 2017] and English Switchboard [Godfrey et al., 1992] corpora. Referring to experimental results the proposed model outperforms the wav2vec2.0 CTC baseline, yielding a WER of 10.6%.

Yang et al (2022), have proposed a model which integrates the Conformer encoder and the wide residual BLSTM network (WRBN) into a fresh Conformer-based acoustic model. This model uses a Conformer encoder, which is built over the Transformers encoder baseline architecture, adding a convolutional network as well as macaron feed-forward layers as described by Gulati et al (2020). The encoder is composed of three main modules: The first module is built over the FFN and is composed of 2 linear layers and several residual connections across it. It applies the utterance-wise Layernorm (LN) on the input and aims to process batches with padding.

The second module is the multi-head self-attention (MHSA) network, which applies positional encoding to encode positional information in the sequence as described by Vaswani et al (2017). In this architecture, it does not apply the scaling in the input signal, but instead, it deviates the positional encoding matrix with a scaling factor.

The third module is built over the convolutional layers, and it applies a pointwise convolution and a GLU. In addition, after a 1-dimensional depthwise convolution and an utterance-wise BN, the Swish activation is applied and finally, a 1-dimensional pointwise convolution is applied (Gulati et al., 2020).

The wide residual convolutional layers (WRCNN), are applied before BLSTM layers and aim to pass the speech representations in a convolution layer. In addition, 3 residual blocks are applied to generate features with different frequencies. To convert these features to the right dimensions, a batch normalization layer and an ELU activation function are applied.

Another innovation that this architecture brings is the replacement of all normalization layers (LN) with utterance-wise LN. In addition, a representation masking is applied during time computation. This improves the accuracy and avoids interference between utterances. Figure 49 shows the full proposed architecture.

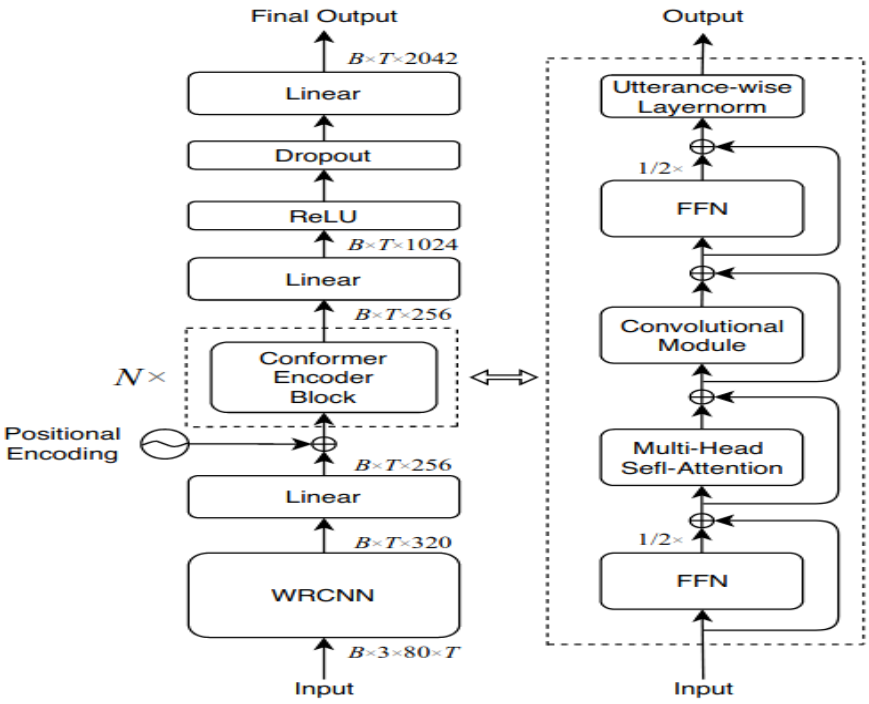

Figure 49. The model architecture of the Conformer-based acoustic model (Wang et al 2022).

The WRCNN and Conformer encoders are combined and employ utterance-wise normalizations. WRCNN processes the input signal, which is then projected to the MHSA dimension by a linear layer. The proposed model is evaluated on the CHiME-4 corpus. It yields a WER of 6.25% and reduces the training time by 79.6%.

Fu et al (2022) have proposed a framework based on LAS - Transformers architecture, which combines the high correlation among speech frames. The three main innovations that this framework brings are:

I. Application of a local attention module to capture the features of the speech sequence.
II. Application of a depthwise convolution layer, to simplify the complexity of the model.
III. Replacement of the absolute positional embedding with relative positional embedding, to improve the representation features.

The full architecture of the proposed model is shown in Figure 50. This architecture is based on the encoder-decoder structure. In the encoder, the speech data first pass to a convolutional layer and then to the encoder layers. Each encoder layer is composed of a local MHA integrated with the relative positional embedding and a FFN. Each sub-layer is composed of a residual structure (He et al., 2016) and a pre-norm mechanism (Xiong et al., 2020). Next, speech features feed the CTC block which outputs the

probability distribution. The decoder is built from the same modules as the encoder, but each layer unlike the encoder contains a masked MHA layer, a MHA layer and a FFN layer.

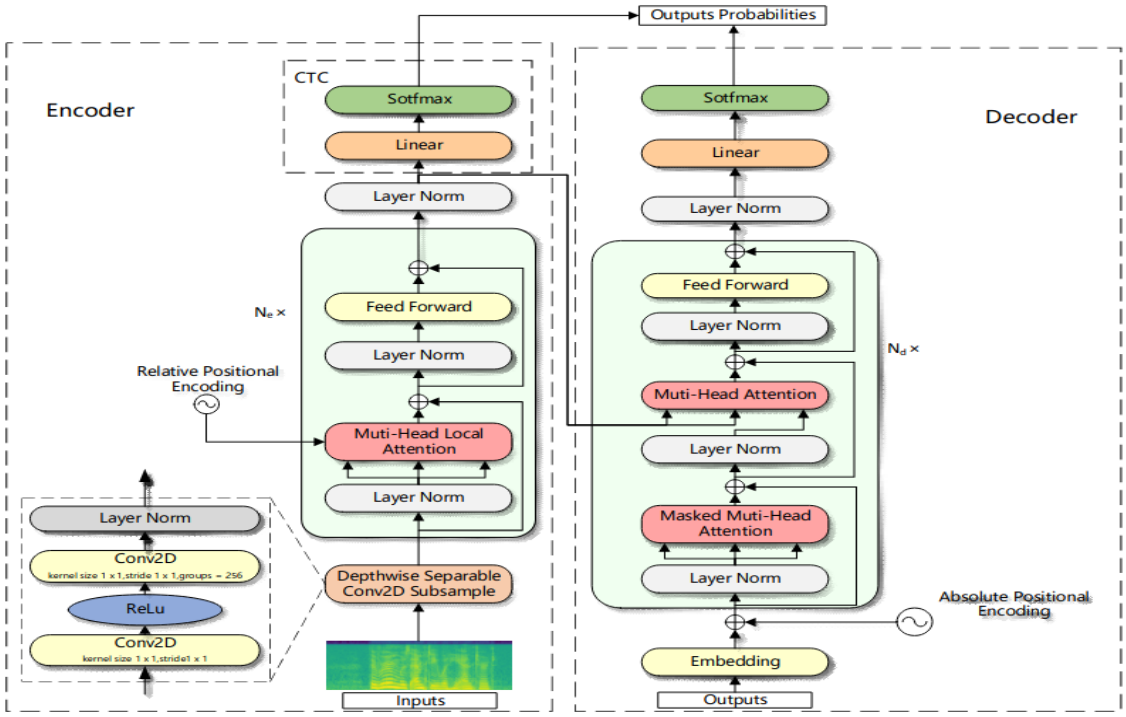

Figure 50. The LAS - Transformers architecture (Fu et al., 2022).

The proposed model is trained and evaluated using the LibriSpeech corpus. Referring to the experimental results, it achieves great results, wherein the test-clean set reaches a WER of 2.2% and in the test-other set it reaches a WER of 5.5%.

The consulted papers in identifying potential advanced end-to-end speech recognition systems have been included in Table 4.

Table 4: Analysis of advanced Transformers-based speech recognition systems.

Authors	Transformers	Cas-Nat	Augment	Jasper	CTC-S2S	Non-Causal-Conv	NAR CTC/attention	WRBN	Conformer	LAS	Wav2vec	BERT	Context	ILS-SSL
Hrinchuk et al (2019)	√			√	√									
Fan et al (2021)	√	√												
Fan et al (2021)	√	√	√											
Shi et al (2021)	√					√								
Deng et al (2022)	√						√							
Yang et al (2022)	√							√	√					
Fu et al (2022)	√									√				
Baevski et al (2020)	√										√			
Baevski et al (2020)	√										√	√		
Mohamed et al (2020)	√												√	
Dong et al (2018)	√				√									
Wang et al (2021)	√													√
Moriya et al (2020)	√				√									

As can be shown from table 4, the topics include different architectures based on Transformers. The baseline architecture of the Transformers is integrated with various approaches such as LAS, Augmentation, CTC, wav2vec etc.

Referring to the analysis that we have done for each paper, each architecture has its advantages and disadvantages. But the aim of our study at this stage is focused on the overall performance of the system, referring to the standard performance measurement parameter word error rate (WER). In Figure 51, we graphically present the value of WER for each architecture.

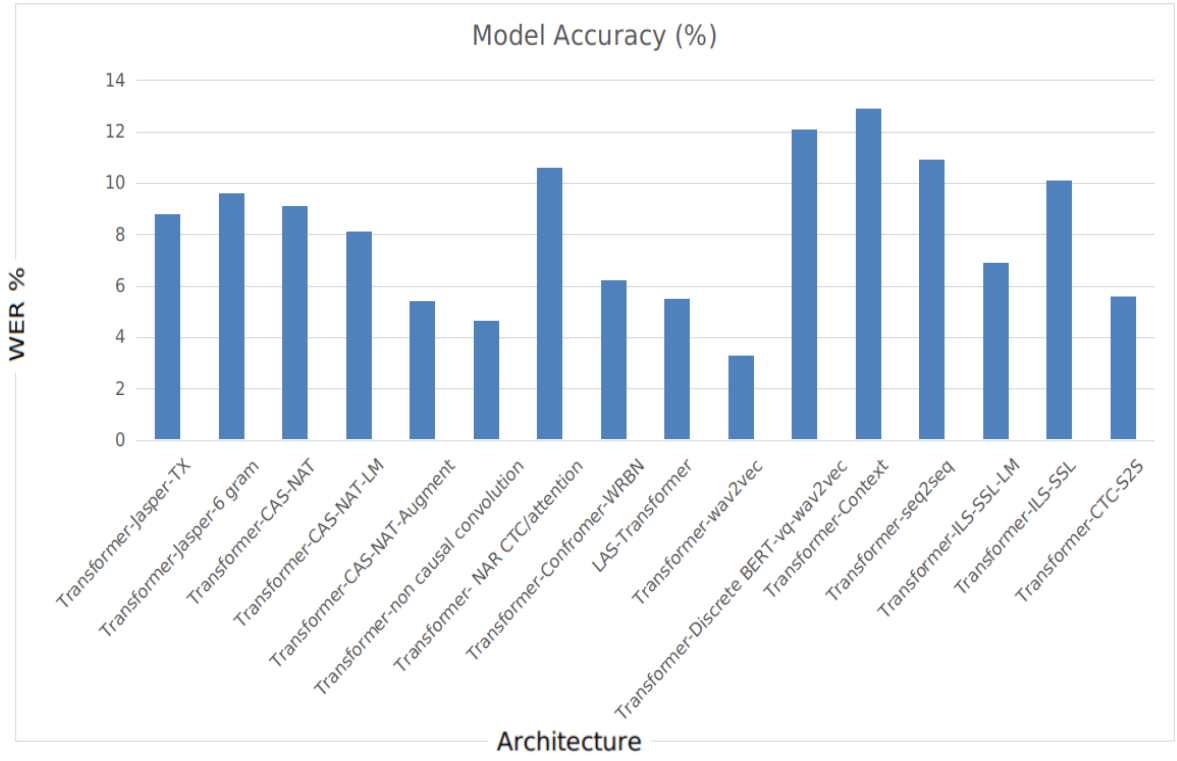

Figure 51. The WER values for all Transformers-based architectures were analyzed.

First, we emphasize that not all analyzed architectures were trained and evaluated with the same corpus. Since corpus size, corpus accuracy, speaker attributes and other corpus parameters are not the same, all these factors jointly directly affect the accuracy of the model. Despite this, the results presented in Figure 51 clearly show which architectures have the best performance. Referring to the presented results in Figure 51, we show that all the architectures analyzed present high performance. We noticed that the architecture that applies the wav2vec approach achieves the best performance. We also emphasize that the application of language models reduces WER by 2 - 5% in some architectures.

2.7 An overview of the Albanian language

The Albanian language belongs to the family of Indo-European languages, which is considered a direct descendant of old Illyrian (Pacarizi, 2008). Today, it is spoken by more than 7 million people in Albania, Kosovo, Macedonia and Montenegro. The Albanian language has 36 phonemes, of which 7 vowel phonemes and 29 consonant phonemes (Orel, 2000). The structure of the word in the Albanian language is divided into smaller units called morphemes (Orel, 2000). Morphemes can be prefixes, suffixes that

serve to form new words, or suffixes to form different forms of the same word. Also, they form nouns, adjectives, verbs, or adverbs and can serve to form the feminine gender of nouns. Names in the Albanian language are divided into specific and general names, spiritual and non-spiritual names, concrete and abstract names, and summary names and subject names. All types of nouns can be singular or plural and are characterized by three genders, masculine, feminine, and neuter (Kadriu, 2010). Names in the Albanian language have five cases: nominative, accusative, genitive, dative, ablative, and vocative; they have 3 forms of inflexion and can be used in both definite and indefinite forms (Opitz, 2006). Albanian verbs are divided into two categories: auxiliary verbs that serve to form the compound forms of the verb and semi-auxiliary verbs that serve to connect the noun part of the subject with the predicate of the subject (Kadriu, 2010). Each verb is presented in one of its six modes: indicative, conjunction, conditional, habitual, desire, and imperative. The three basic tenses that characterize the verbs in Albanian are the present, the past, and the future. Another important class in the Albanian language is adverbs which complement the verb indicating time, manner, place, and quantity (Güçlü, 2015). For the construction of sentences in the Albanian language, we have two main elements which are the verbal group and the noun group. The noun group has several forms and can be formed mainly from a noun and a determiner or numeral, from a noun and an adjective, from the union of two nouns that may belong to different races, and from a noun and an adverb. The noun group represents the subject of the sentence. Like the noun group, the verb group has several forms, it can be formed from a transitive verb and an adverb, only from a verb without an adverb, from a verb, adverb, and a circumstance, from an auxiliary verb and a noun or adjective group. The Albanian language has two main dialects geographically divided into two larger regions; the northern dialect or Geg and the southern dialect or Tosk. Geg speakers are about two-thirds of Albanian speakers. The main differences between them are phonetic, but there are also some grammatical differences, mainly of a morphological nature. Meanwhile, changes in syntax are almost negligible. Differences in the lexicon in supplementary forms are conditioned by the conditions and circumstances in which the inhabitants of these dialectal areas of the same language lived.

2.8 Conclusions

In this section, we presented a systematic literature review to provide a comprehensive understanding of speech recognition technology. We have analyzed the most cited and popular ASR systems during the last decade targeted towards 1) hybrid speech recognition systems; 2) end-to-end speech recognition systems; 3) ASR systems for low-resource language, and 4) Transformers-based ASR systems. For each selected architecture, we have made a detailed analysis, describing all the components with which it is built. In addition, we have analyzed the performance of each architecture by referring to the international standard word error rate (WER) parameter. At the end of this session, we have defined the best architecture, which we can use as a good reference to design ASR systems for the Albanian language. Referring to the analyzed performance that we have done for each architecture we conclude that the end-to-end models and Transformers-based models achieve state-of-the-art performance. In addition, hybrid architectures when integrated with Transformers and applying a language model give great results. While for low-resource ASR systems, end-to-end-based architecture and Transformers-based architecture achieve state-of-the-art performance.

3. METHODOLOGY

In this chapter, we present the analytical methods to achieve the set objectives in the thesis, by describing the model design, corpus development, tools and assessment criteria. The proposed models aim to fulfil the objectives of our study and to answer the hypotheses and research questions targeted. The main goal of the thesis is to design a framework for Albanian Speech Recognition, to be able to recognize general Albanian speech produced by any speakers with state-of-the-art performance.

3.1 Model design

After a deep analysis that we have done in chapter 2, where we analysed more than 150 papers in the domain of speech recognition, we concluded that end-to-end models and Transformers-based models achieved state-of-the-art performance, specifically for low-resource languages. For the Albanian language, we will design two baseline architectures. The first architecture will be based on end-to-end approaches, while the second will be based on Transformers-based approaches.

3.1.1 End-to-end ASR for the Albanian language

In this session, we describe the proposed model based on end-to-end approaches. It is composed of two main modules: Residual Convolutional Neural Networks (ResCNN) (Vydana and Vuppala 2017) and Bidirectional Recurrent Neural Network (BiRNN) (Kamath et al. 2019). The ResNets are composed of stacking several residual blocks that enable the direct connection of the lower layers with the higher layers on the network. Residual learning is proposed to overcome degradation problems. In addition, it reduces the training time of the model and improves the accuracy in the depth of the network. The BiRNN consists of a forward RNN layer and a backward RNN layer, by considering both the past and the future features to make predictions. The BiRNN is constructed of a version of RNNs called GRUs as described by Dey and Salem (2017). It combines long and short-term memory into its hidden state. It consists of a hidden state, the reset gate and the update gate. The function of the update gate is to know how much of the passed memory to retain, while the reset gate knows how much of the memory to forget. The GRU requires fewer computational resources than baseline RNNs. In Figure 52, we have presented the proposed model in detail. The audio waves generated by the corpus are transformed into Mel Spectrograms features, which feed the ResCNNs Layers. The ResCNNs aim to learn referenced

residual functions by applying skip connections. The output of ResCNNs feeds the BiRNNs, which aims to leverage the learned ResCNNs audio features.

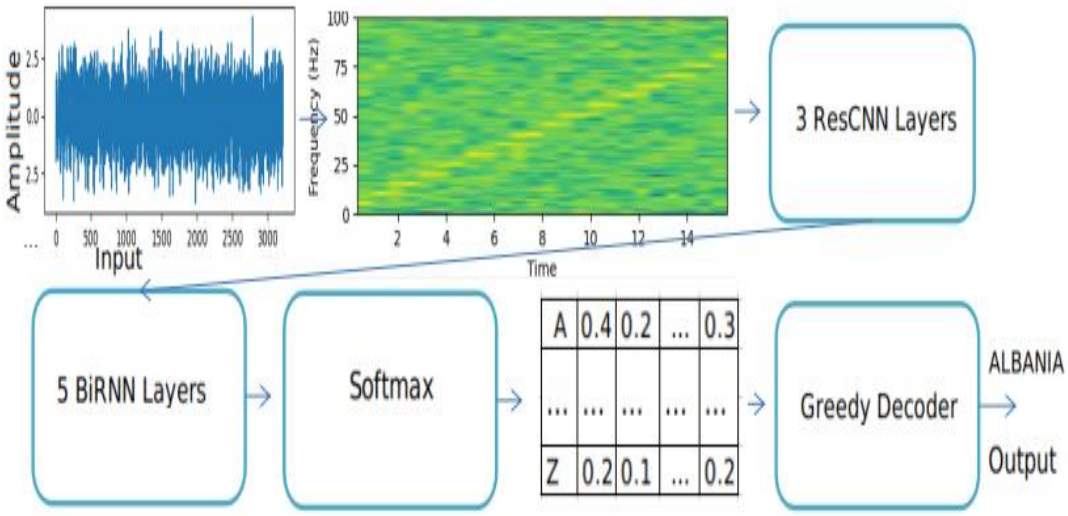

Figure 52. Proposed end-to-end ASR system for the Albanian language.

Another very important module in our architecture is Connectionist Temporal Classification (CTC) (Graves et al., 2006). It is a type of neural network output for training recurrent neural networks (RNN) to tackle sequence problems where the timing is variable. The CTC algorithm applies a blank token that acts as a null delimiter. This token is removed after collapsing repeated predictions, allowing repeated sequences and periods of “silence”. In this way, the blank token is not included in the loss computation. It allows a direct alignment between the input and the output without forcing a classification of the output vocabulary.

The next module in this architecture is Softmax (Liu et al., 2016). It is a mathematical function used to normalize the output of BiRNN to a matrix probability distribution. And the last module of the architecture is Greedy Decoder (Battenberg, 2017). It is fed with the matrix probability distribution and outputs the transcript. In addition, we have applied the AdamW optimizer (Loshchilov and Hutter, 2017), which accelerates the convergence time of the model and also significantly reduces the compute time.

3.1.2 Transformers-based ASR system for the Albanian language

In this session, we describe the proposed model based on Transformers. This model is based on Wav2Vec2 2.0 baseline architecture as described by Baevski et al (2020). The wav2vec 2.0 masks the input row audio waves by converting them into compressed representations and applies self-supervised learning over a quantization of compressed representations without labels. Figure 53 shows in detail the proposed architecture.

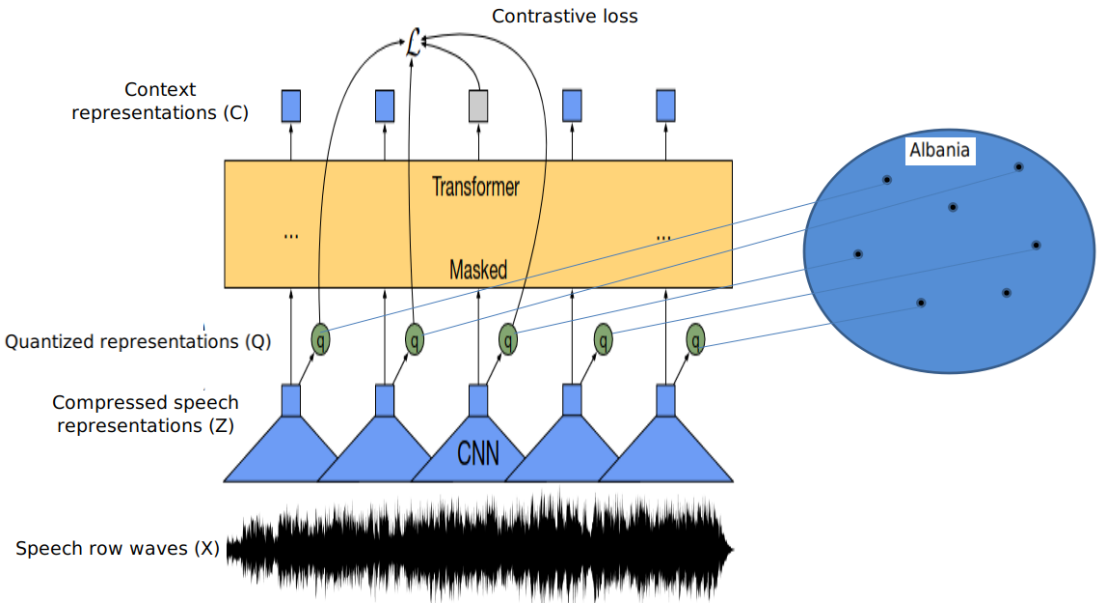

Figure 53. Proposed Transformers-based ASR system for the Albanian language.

Initially, we apply a multi-layer CNN (encoder), which transforms input audio wave features $(x_1, x_2, \ldots\ldots\ldots, x_n)$ into latent speech representations, which means a sequence of vectors$(z_1, z_2, \ldots\ldots, z_n)$. The encoder is composed of some blocks which contain a 1-D fully convolutional network (1D FCN) integrated with causal convolutions as described by Bai et al (2018), followed by a normalization layer and a RELU activation function.

Then, the latent speech representations are discretized into a certain group by applying a quantization module (Jegou et al, 2011). We apply the Gumbel-Softmax distributions to choose an approximate discrete group (Jang et al., 2016). Next, we apply a spectral masking approach to the quantized representations (Park et al, 2019).

The masked quantized features feed the Transformers which add features to the entire speech sequence and construct the context representations$(c_1, c_2, \ldots\ldots\ldots c_T)$, as described by Baevski et al (2020) and Delvin et al (2018). Taking into account that the masked quantized representations sequence is too long, we have applied a relative positional which enables the Transformers to know the order of the sequence.

The training of the model goes through two phases: 1) pre-trained model, and 2) fine-tuned model.

During the pre-training phase, the model learns representations of row audio waves which are identified as the correct masked quantized speech representation for a time step.

One of the innovations that this architecture brings is the application of two loss functions. The first is the Contrastive Loss function (Oord et al., 2018), which extracts useful masked quantized latent speech representation for each time step. And the second is Diversity Loss (Dieleman et al., 2018), which increases the use of the quantized group representations. And the total loss will be $L = L_m + L_d$ as shown in Figure 53.

During the second phase, the pre-trained model is fine-tuned by adding a randomly initialized linear projection to the context representations. In addition, we apply a CTC loss function (Graves et al, 2006) and a Spec Augment technique (Park et al., 2019) to avoid overfitting and reduce WER for CASR corpus.

3.2 Corpus development

In this section, we report corpus design. During its construction, we have considered the features of the Albanian language as well as attributes such as phonetics, adverse environment conditions (clean, noisy); speaker attributes such as; age, gender, accents, speed of utterance, dialects; training process and voice recording device. ASR systems are developed to work with short audio segments and perfectly transcribed reference texts. Hence, we have prepared the data into a usable state for ASR systems. This section describes data collection, audio and text processing as well as corpus description and organization procedure.

3.2.1 Source Data

To build the Albanian corpus (CASR) we have obtained 100 hours of audio recordings from 200 audiobooks that belong to various topics like biography, social and political sciences, psychology, religion, economics and business, history, philosophy and sociology. Audiobooks are recorded in MP3 format, using 32-bit float compression at 16 kHz and they are in both dialects of the Albanian language (tosk and geg). Speakers belong to different age groups ranging from 20 to 70 years old so that the corpus is as heterogeneous as possible. We have selected this type of raw data to convert into a corpus suitable for Albanian ASR because they present a low presence of noise in the audio recording and small compression loss. The audio recordings are transcribed strictly verbatim, listening carefully several times to the speeches.

3.2.2 Audio and Text Preprocessing

The audio recordings selected to create the corpus have different lengths ranging from 1 to 3 hours. Taking into account the fact that acoustic models are trained with short audio recordings, the first step we have taken to build the corpus is to cut the audio recordings into short sequences ranging from 2 to 12 seconds. For this purpose, we have used the Audacity tool (Audacity, 2017). Each audio recording is converted from mp3 to flac format using a 16-bit linear PCM sample encoding (PCM_S16LE) sampled at 22.05 kHz and is exported to the Audacity tool, where it is divided into short sequences. Then, each audio recording sequence called utterance is named with a four-digit decimal number randomly. In the case of audio sequences where the pronunciation between words is too long, we have cut them leaving no more than 2 seconds in length. After we cut all the audio recordings, in total we created 37758 utterances. Each utterance called an audio file has an average of 20 words.

Regarding the creation of text files, we have listened carefully several times to each audio file, and we have written strictly verbatim the corresponding transcripts for each utterance. All transcripts are normalized by converting them into upper-case, removing the punctuation, and expanding common abbreviations. In this form, we have created the audio and text files of the CASR corpus.

3.2.3 Corpus description and organization

After creating all the audio files and text files which are saved in formats (.flac) and (.text), we have organized them according to the objectives of this study. The final corpus has a size of 12.3 GB with a total duration of speech data of 100 hours. To evaluate the impact of corpus size on the accuracy of the model, and to help the researchers who want to train and test their models, overcome hardware limitations, we have created some subsets. In Table 5 we have reported the size in hours, the number of utterances, the number of words, and the average length of utterance for CASR and each subset created.

Table 5: Characteristics of the CASR and its subsets.

Corpus	**Subset**	**Size**	**Number of utterances**	**Total words**	**Avg. length per utterance**
CASR	Whole	100 h	37 758	848 553	9.53 s
CASR_1	Train	80 h	29 215	686 971	9.85 s
	Test	20 h	8 543	161 582	8.42 s
CASR_2	Train	40 h	15 652	339 831	9.20 s
	Test	10 h	3 727	85 155	9.65 s
CASR_3	Train	8 h	3 520	68 144	8.18 s
	Test	2 h	784	17 211	9.18 s
CASR_4	Train	48 min	289	5 423	9.96 s
	Test	12 min	75	1 531	9.60s

3.3 Hardware Setup

One of the biggest challenges we faced during our study was the hardware setup. We pick five PCs to perform the experiments, since the resources we have at hand, especially the number and power of GPUs are limited. Each PC used an Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz 3.19 GHz as a CPU; 16.0 GB and 1 TB Hard Disk. The corpus was stored on each PC to allow for a faster data stream during training. Each PC used a NVIDIA GeForce GT 710 as GPU. Each PC performs independently with a model set by us. The models have been trained for approximately one month until they have converged.

3.4 Assessment Criteria

3.4.1 Word Error and Word Error Rate

Word error is calculated by computing the Levenshtein distance between the reference sequence and hypothesis sequence at the word level. Levenshtein distance is a string metric for measuring the difference between two sequences (Fiscus et al., 2006). Informally, the Levenshtein distance is defined as the minimum number of single-character edits (substitutions, insertions or deletions) required to change one word into the other. The edits to word level can be naturally extended when calculating Levenshtein distance for two sentences. While word error rate (WER) compares reference text and hypothesis text at the word level (Morris et al., 2004). Mathematically WER is defined as:

$$\mathrm{WER} = \frac{\mathrm{S_w} + \mathrm{D_w} + \mathrm{I_w}}{\mathrm{N_w}}$$

Where:

- $\mathrm{S_w}$ is the number of words substituted
- $\mathrm{D_w}$ is the number of words deleted
- $\mathrm{I_w}$ is the number of words inserted
- $\mathrm{N_w}$ is the number of words in the reference

3.4.2 Character Error and Character Error Rates

Character error is calculated by computing the Levenshtein distance between the reference sequence and hypothesis sequence at the character level. While character error rate (CER) compares the reference text and hypothesis text in the char-level (Chung et al., 2017). Mathematically CER is defined as:

$$\mathrm{CER} = \frac{\mathrm{S_c} + \mathrm{D_c} + \mathrm{I_c}}{\mathrm{N_c}}$$

Where:

- $\mathrm{S_c}$ is the number of characters substituted
- $\mathrm{D_c}$ is the number of characters deleted
- $\mathrm{I_c}$ is the number of characters inserted
- $\mathrm{N_c}$ is the number of characters in the reference

3.5 Conclusion

In Chapter 3, we presented the proposed models for Albanian ASR, as well as corpus development, tools and assessment criteria.

In Section 3.1, the end-to-end proposed model and all its components have been presented. In addition, the Transformers-based ASR system for the Albanian language has been presented.

Section 3.2 presents the corpus development. It includes source data, audio and text preprocessing and corpus description and organization. Hardware configurations are presented in Section 3.3.

Finally, in Section 3.4 the model evaluation parameters word error rate (WER) and character error rate (CER) is presented.

4. EXPERIMENTS AND RESULTS

In this chapter, we present all the experiments and results to support our research.

First, we conduct an analysis of various architectures based on end-to-end approaches. Initially, we train the model with the training set to find the optimal network layer size, and to define model parameters as well as its training time. Referring to the results of the experiments, we select the best architecture to continue with other experiments according to the objectives of the study. In addition, we evaluate the effect of corpus size, voice and dialect on the accuracy of the model.

Also, we compare the CASR corpus with the LibriSpeech corpus in terms of WER and CER. Second, we conduct an analysis of Transformers-based architecture by training them with CASR corpus. In addition, we conduct a comparison of the proposed end-to-end architectures with the Transformers-based architecture.

4.1 Evaluation of different architectures through the training set.

For a well-trained ASR system, it is imperative to choose the proper model capacity. If the model is too simple with a few layers it will not perform well due to over-fitting problems. On the other hand, if the model becomes highly complex, it requires a lot of time to be trained and often adverse effects in performance may be observed as it cannot generalize and recognize unseen speech data. To address this issue, we have conducted a study on the relation between the number of RNN and GRU layers and the accuracy of the model. We have gradually increased model capacity starting from one RNN layer to three layers and from two GRU layers to five layers. In total, we have evaluated five different architectures. The first architecture has 3 residual CNN layers and 5 Bidirectional GRU layers; the second architecture has 1 residual CNN layer and 4 Bidirectional GRU layers; the third architecture has 1 residual CNN layer and 3 Bidirectional GRU layers; the fourth architecture has 1 residual CNN layer and 2 Bidirectional GRU, and fifth architecture has 2 residual CNN layer and 2 Bidirectional GRU.

All architectures are executed at the same time, on five separate PCs that have the same GPU, CPU and memory. They are trained on the validation set (whole corpus), which contains 100 hours of speech data with their transcripts. All experiments are done in the same conditions related to hyper-parameters as shown in Table 6

Table 6. Hyper-parameters and experiment settings

Hyper-parameters	Values
Input Features	Mel-Spectrogram
Learning-rate	5e - 4
Bach size	10
Epoch	100
RNN layers	1, 2, 3
CNN layers	2, 3, 4, 5
dropout	0.1

Figure 54 shows the relation between the number of RNN and GRU layers related to WER and CER.

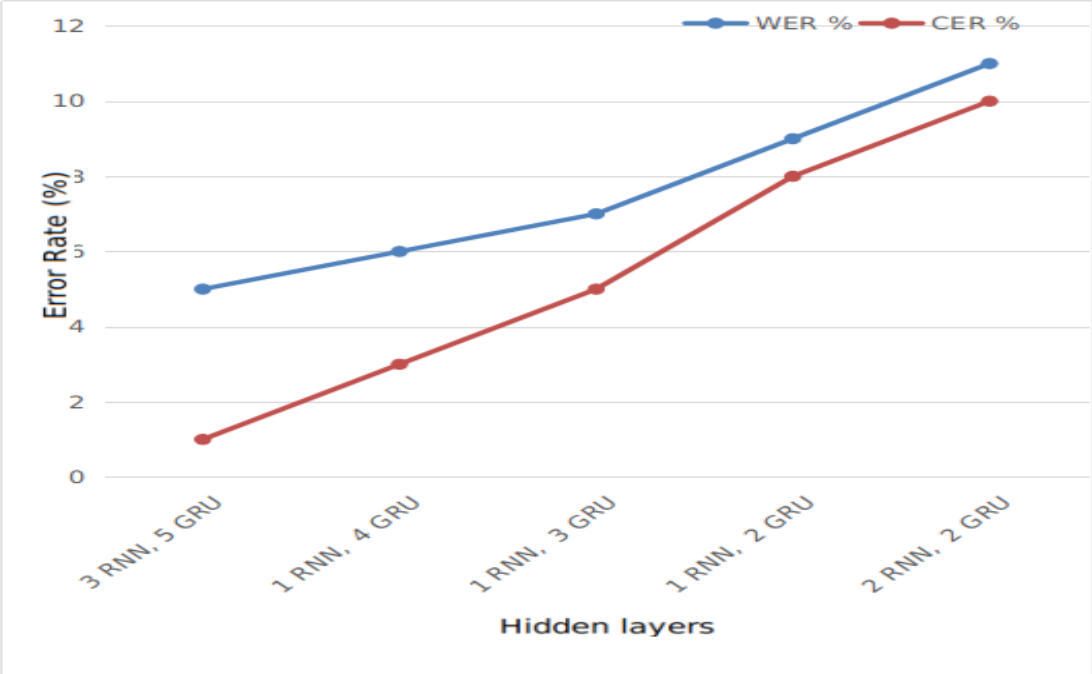

Figure 54. Impact of number of RNN and GRU layers on WER and CER.

We observed that the acoustic model with three RNN layers and five GRU layers outperformed the rest of the architectures achieving a WER of 5% and a CER of 1% on the training set. Also, we observed that with increasing the number of layers the performance of ASR can be significantly improved. However,

we faced difficulties as we were increasing the number of layers beyond 3 RNN and 5 GRU due to increased training time and the unavailability of computational resources.

In Table 7, we have reported training time in hours of each architecture measured for 100 epochs

Table 7. Training time

Architectures	Training time in hours
3 RNN, 5 GRU	1216 hours
1 RNN, 4 GRU	967 hours
1 RNN, 3 GRU	722 hours
1 RNN, 2 GRU	502 hours
2 RNN, 2 GRU	915 hours

Taking into account the performance of the model and its training time, we have selected the model with 1-layer RNN and 3 layers GRU as the most suitable for the continuation of our experiments.

4.2 Evaluation of proposed model through the testing set.

After evaluating different architectures as we described in the session above, we should end up with an evaluation of the final model. For this purpose, we have created a training set and a testing set with a split ratio of 80:20. Which means that 80 hours are used for training and 20 hours are used for testing (See Table 5). The selection of data was done randomly, and both subsets include speakers of the age group from 20 to 70 years old, both genders, and use both Tosk and Geg dialects of the Albanian language. The training set is always bigger than the other sets. In general, this data set is used to evaluate the parameters of the model and the depth of the network in terms of layer size. The testing set is used to evaluate the accuracy of the model for unknown data. In other words, through the testing set, we evaluate whether a model can be generalized even for unknown data. In Figure 55, we have reported the results of experiments for both WER and CER related to the number of epochs.

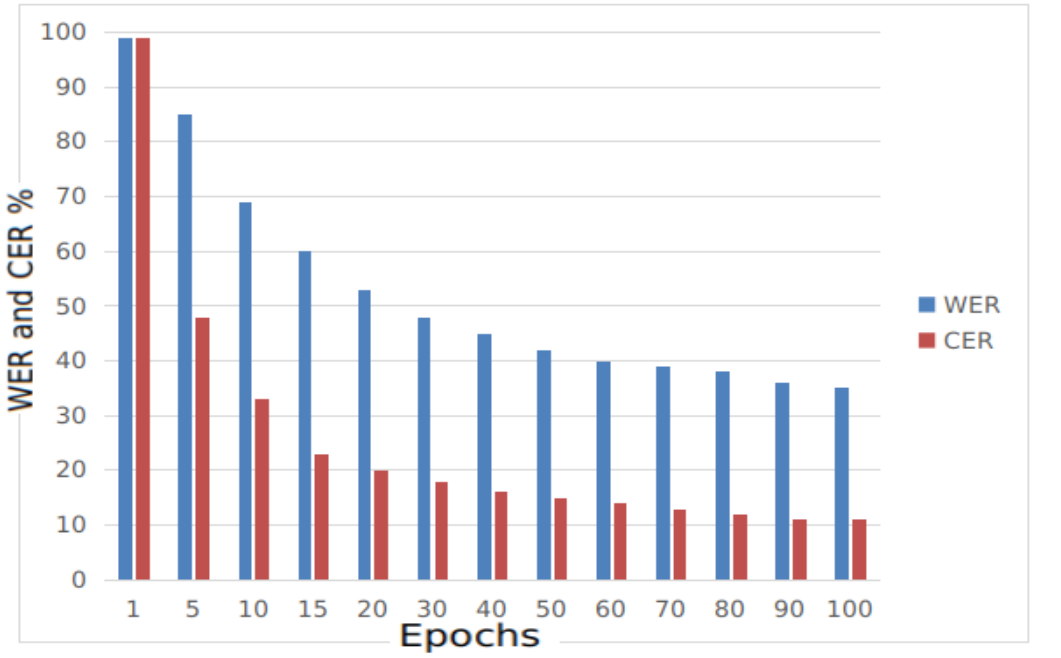

Figure 55. The performance on the testing set in terms of WER and CER

As can be shown in Figure 55, CASR has yielded 11% CER and 35% WER on the testing sets. Note that after epoch 50, the curves start to become linear for both the case of WER and the case of CER. By analyzing the results obtained and comparing them with the results analyzed during the literature study, we conclude that end-to-end deep learning techniques are suitable for Albanian ASR.

4.3 Evaluation of the effect of corpus size on the accuracy of the model.

To address this issue, we have conducted some experiments on the relation between the size of the corpus and WER and CER metrics. For this purpose, we have created four subsets with 1 hour, 10 hours, 50 hours and 100 hours as described in Table 5. Evaluation is done on the testing set and the experiments have continued up to 100 epochs. Each subset is divided into a training set and a testing set with a ratio of 80:20 randomly, which means 80% of the corpus is used to train the model and 20% is used to test it (See Table 5). Even in this case, the splitting of data was done randomly, and both subsets (training set and testing set) include speakers of the age group from 20 to 70 years old, both genders, and use both Tosk and Geg dialects of the Albanian language. All experiments are done with the same hyper-

parameters and the architecture is composed of 1 layer of RNN and 3 layers of GRU. Figure 56 shows the results of the experiments.

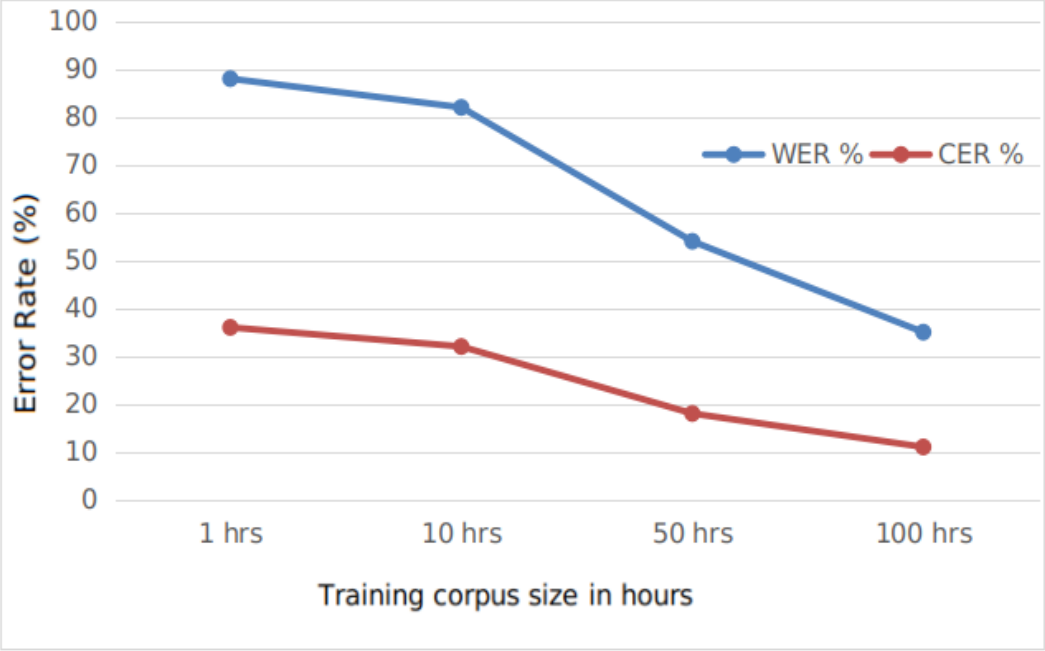

Figure 56. Effect of training corpus size on WER and CER with CASR.

In the first case, when the model is trained with a corpus of one hour of which 48 minutes were used for training and 12 minutes for testing, it yielded 36% CER and 88% WER on its testing set. The results obtained for this corpus size are unacceptable, which means that to build and evaluate an ASR model, the corpus size of one hour is insufficient.

In the second case, when the model is trained for 10 hours of which 8 hours were used for training and 2 hours for testing, it yielded 32% CER and 82% WER on its testing set. Referring to the WER and CER values obtained for this case, we conclude that the corpus size of ten hours is insufficient to build and evaluate an ASR model.

In the third case, when the model is trained with a corpus of fifty hours of which 40 hours were used for training and 10 hours for testing, it yielded 18% CER and 54% WER on its testing set. Even in this case, the size of the corpus is insufficient to build and evaluate an ASR model, as the value of WER is too big. In the last case when the model is trained with a corpus of hundred hours of which 80 hours were used

for training and 20 hours for testing, it yielded 11% CER and 35% WER on its testing set. In this case, the values of WER and CER are considered acceptable, compared to the results analyzed in the literature.

As can be shown even from Figure 56, with increasing corpus size we have obtained a significant decrease in WER and CER. It means that the size of the corpus affects the accuracy of the model. Large corpora give better results of CER and WER and converge faster in terms of accuracy. Due to limited resources, we could not create a larger corpus to show further how the value of WER and CER would depend on a further increase in corpus size. But, we emphasize once again, that with the corpus of 100 hours we have received satisfactory results for WER and CER, which are comparable to the results in the literature.

4.4 Evaluation of the effect of the voice and dialect of the speaker on the accuracy of the model.

To address this issue, we have created two subsets, each with 20 hours, called Clean_CASR and Mix_CASR. Clean_CASR is based on bible audiobooks, where transcripts and audios are in the Albanian standard language. It includes only one male speaker that speaks in standard Albanian language without using dialects. The topics are concerned with religious issues using a rich vocabulary. Mix_CASR is based on 200 audiobooks that belong to various topics like biography, social and political sciences, psychology, religion, economics and business, history, philosophy and sociology. It includes some speakers that speak in standard Albanian language and both Tosk and Geg dialects. The speakers belong to both genders and different age groups from 20 to 70 years old. For both subsets created, we use the same architecture that is composed of 1 RNN layer and 3 GRU layers.

All hyper-parameters are the same for both cases, as they are described in Table 6. Evaluation is conducted for 100 epochs. In Figure 57, we have reported the WER results for both subsets.

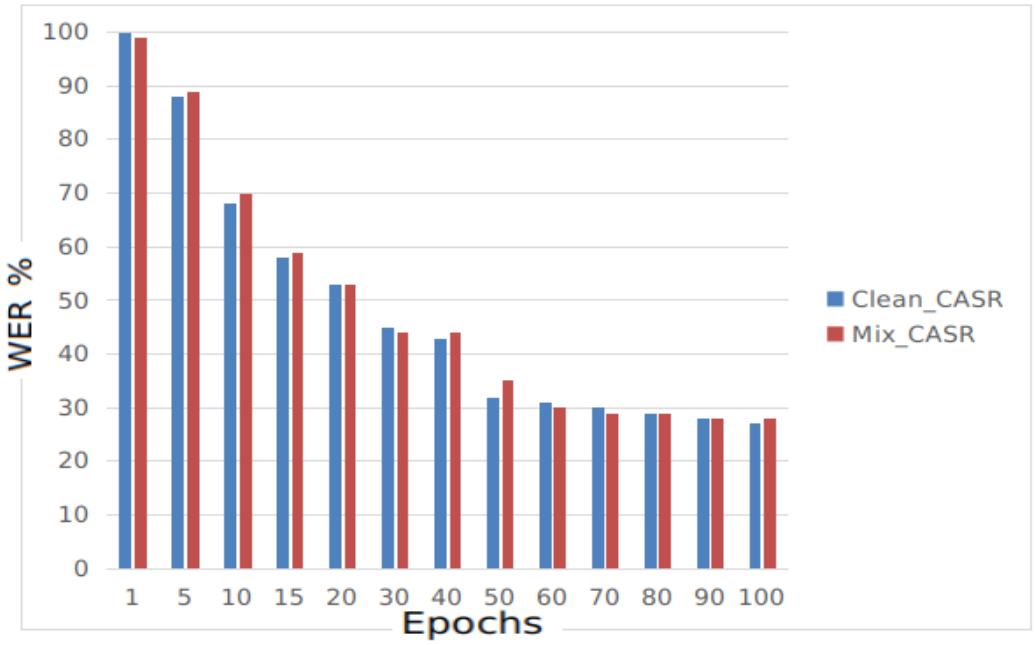

Figure 57. The performance on the training set in terms of WER.

In this case, when the model is trained with Clean_CASR corpus, it has yielded 27 % WER on its training set. While, in this case when the model is trained with Mix_CASR corpus, it has yielded 28 % WER on its training set. Even in these cases, it is noticed that the curves start to become linear after epoch 50, undergoing very small changes as well as following each other.

In Figure 58, we have reported the CER results for both subsets.

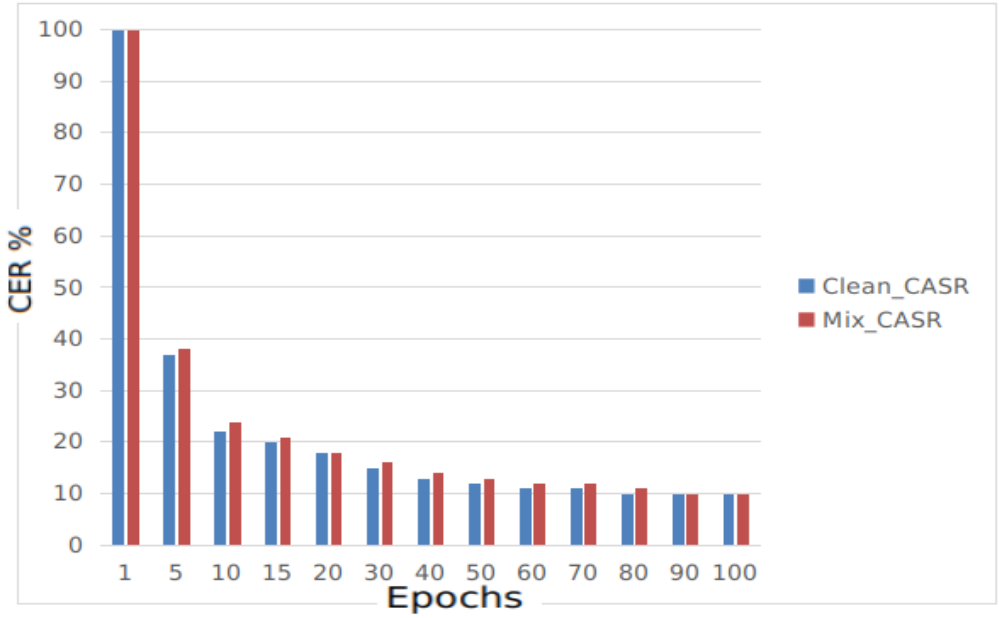

Figure 58. The performance on the training set in terms of CER.

In this case, when the model is trained with Clean_CASR corpus as well as when the model is trained with Mix_CASR, it has yielded 10 % CER on its training set. As shown even in Figure 58, the curves start to become linear after epoch 50, undergoing very small changes as well as following each other.

4.5 Evaluation of CASR in comparison to LibriSpeech.

To address this issue, the acoustic model is trained first with CASR and then with LibriSpeech corpora. In both cases, we have created a training set and a testing set with a split ratio of 80:20 randomly, specifically the training set of 80 hours and the testing set of 20 hours. All hyper-parameters are the same for both AMs (See Table 6). For both cases, we have chosen the architecture with 1 layer RNN and 3 layers GRU, and we have trained and tested both with 100 epochs. In Figure 59, we have reported the results of experiments for WER related to the number of epochs for both CASR and LibriSpeech corpora. In the case when the model is trained with the CASR corpus, it has yielded 35% WER on its own test set. While, in the case when the model is trained with the LibriSpeech corpus, it has yielded 32% WER on its own test set. Even though WER has a lower value of 3% in the case when the model is trained with LIbriSpeech, both cases give satisfactory results. As shown in Figure 59, as the number of epochs increases the curves start to fall, by following each other. Also, it is noticed that after epoch 50 curves start to become linear.

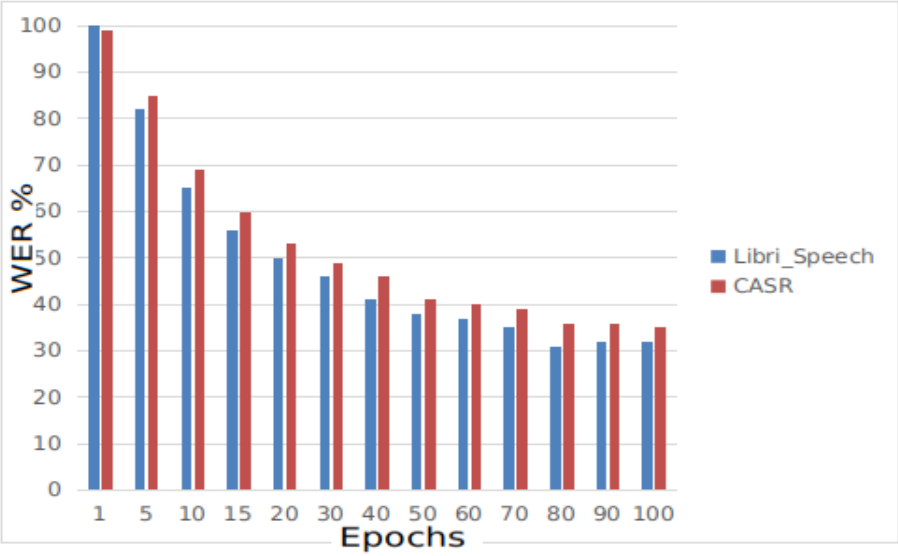

Figure 59. Experiment results on the relation between the WER and epochs.

While in Figure 60 we have reported the results of experiments for CER depending on the number of epochs for both CASR and LibriSpeech corpora. In the case when the model is trained with the CASR corpus, it has yielded 11 % CER on its own test set. While, in the case when the model is trained with the LibriSpeech corpus, it has yielded 9 % CER on its own test set. As can be shown from Figure 60, both cases present almost the same CER results. The model trained with LibriSpeech gives a CER of 2% lower than the model trained with CASR, which is a negligible value. Even with this parameter (CER) after epoch 50 the curves start to linearize, undergoing very small changes.

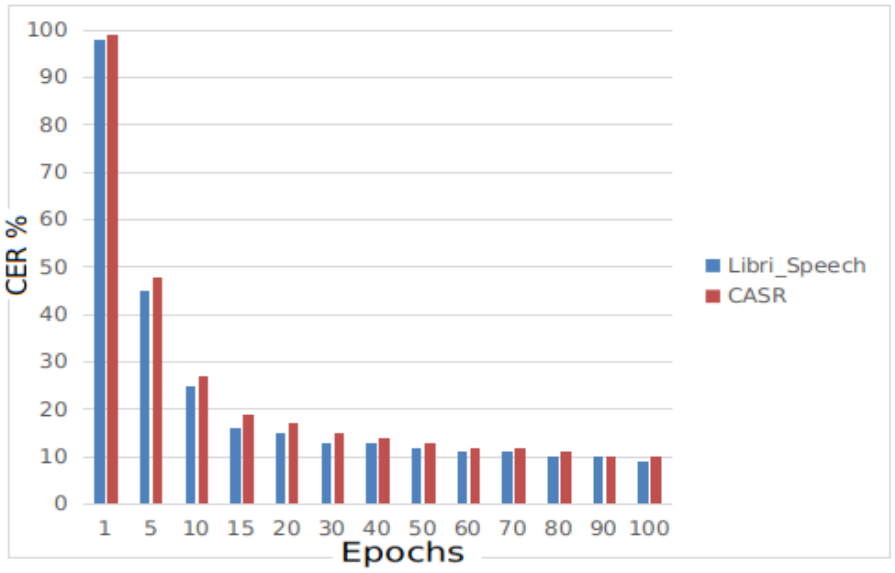

Figure 60. Experiment results on the relation between the CER and epochs.

4.6 Evaluation of Transformers architecture using CASR corpus

In this section, we present the experiments and results for the proposed Transformers-based architecture. To train the model we used training sets, which consist of 80 hours of audio recording along with their transcripts. While to test the model we used testing sets, which consist of 20 hours of the audio recording along with their transcripts. Splitting of data on both training and a testing subset is done randomly (See Table 5). Audio recordings are made from some speakers that speak in standard Albanian language and both Tosk and Geg dialects. The speakers belong to both genders and different

age groups from 20 to 70 years old. In table 8 we have reported all hyper-parameters that were used during training.

Table 8. All hyper-parameters

Hyper-parameters	Values
Learning_rate	0.0002
Train_batch_size	18
Eval_batch_size	18
Seed	42
Gradient_accumulation_steps	2
Total_train_batch_size	16
Optimizer	Adam with betas=(0.9,0.999) and epsilon=1e-08
lr_scheduler_type	linear
lr_scheduler_warmup_steps	500
Num_epochs:	10
Mixed_precision_training	Native AMP

In Table 9, we have reported experimental results. Referring to the WER parameter which is the main indicator of the performance of an ASR system, it has reached 18 %. This result is impressive for the ASR in the Albanian language. Also, from the experimental results, it is noticed that with Transformers the model converges quickly. Up to epoch number 10, we have received the minimum of WER. After epoch 10, the results undergo negligible change.

Table 9. Training results

Training Loss	Epoch	Step	Validation Loss	WER
5.16	0.1	200	29.123	0.9707
0.7994	0.5	1000	0.3889	0.3773
0.6853	1.0	2000	0.3244	0.2906
0.6491	1.5	3000	0.2961	0.2576
0.6154	2.0	4000	0.2861	0.2424
0.5978	2.5	5000	0.2855	0.2364
0.5578	3.0	6000	0.2685	0.2243
0.5375	3.5	7000	0.2701	0.2197
0.5299	4.0	8000	0.2632	0.2081
0.4795	4.5	9000	0.2653	0.2072
0.5119	5.0	10000	0.2672	0.2089
0.4785	5.5	11000	0.2653	0.2004
0.4563	6.0	12000	0.2604	0.1997
0.4424	6.5	13000	0.2674	0.1963
0.4472	7.0	14000	0.2686	0.1942
0.4352	7.5	15000	0.2764	0.1934
0.419	8.0	16000	0.2789	0.1909
0.3963	8.5	17000	0.2835	0.1889
0.4067	9.0	18000	0.2838	0.1887
0.3894	9.5	19000	0.2896	0.1884
0.3758	10.0	20000	0.2893	0.1863

A very important parameter in evaluating the performance of a model is also the training time. The proposed Transformers-based model needs about 3 days to train and evaluate the model.

4.7 Conclusion

In Chapter 4, we presented all the experiments and results to support our research.

In Section 4.1, we analysed various architectures based on end-to-end approaches. We trained and evaluated the proposed models with the training set to find the optimal network layer size, and define model parameters and training time. We selected the best architecture to continue with other experiments according to the objectives of the study.

Section 4.2 presents the evaluation of the best-selected architecture through the testing set.

Sections 4.3 and 4.4 present the evaluation of the effect of corpus size, voice and dialect on the accuracy of the model according to the hypotheses addressed.

Section 4.5 presents a comparison of the CASR corpus with the LibriSpeech corpus in terms of WER and CER.

Finally, Section 4.6 presents an analysis of Transformers-based architecture by training them with CASR corpus. In addition, a comparison of the proposed end-to-end architectures with the Transformers-based architecture is presented.

5. DISCUSSION OF FINDINGS

In this thesis, we investigated various architectures based on deep learning techniques for speech recognition of the Albanian language. In addition, we designed a corpus for the Albanian language, which contains 100 hours of audio recording and their transcripts, used to train and evaluate various ASR architectures. This research aimed to create a framework for Albanian speech recognition. The findings of this thesis are classified into two categories: 1) secondary findings derived from the literature review, and 2) primary findings derived from our experiments.

5.1 Secondary findings

The literature review aimed to present a comprehensive understanding of speech recognition technologies by analyzing various architectures with all components. The main goal of LR was to identify the best architecture so that we can use it as a reference to design ASR models for the Albanian language. To accomplish these objectives the appropriate research papers have been dealt with. We classified these papers into four categories: 1) hybrid ASR systems; 2) end-to-end ASR systems; 3) ASR systems for low-resource language, and 4) Transformers-based ASR systems.

5.1.1 Hybrid ASR systems

Hybrid ASR systems are mainly built by integrating classic HMM and GMM models with deep learning approaches and recently with Transformers technology. As is shown in Table 1, we analyzed the most popular and cited hybrid architectures during the last decade. The first hybrid ASR systems were mainly built by integrating classical HMM, and GMM models with DNN approaches. Then, with the development of recurrent neural networks (RNN) and different approaches in the applications of ASR systems, traditional hybrid systems based on commercial HMM-GMM models moved to hybrid models combined with RNN and their versions. And today, with the development of Transformers and their application in speech recognition, hybrid systems integrate Transformers, classical models HMM, GMM, DNN, and different approaches based on RNN into a single system. The performance of each architecture was analyzed based on the international standard word error rate (WER) parameter. Regardless that the training and evaluation of the models analyzed were not done with the same corpus, which means that corpus size, corpus accuracy, speaker attributes etc., are not the same which directly affects the value

of WER, the hybrid systems which integrate Transformers, classical models HMM, GMM, DNN, and different approaches based on RNN achieve state-of-the-art WER results (See Figure 10).

5.1.2 End-to-end ASR systems

The end-to-end ASR systems integrate the acoustic model, language model and pronunciation model into a neural network. Integration of all these components jointly provides a simple training process, reduces the training and decoding time, as well as improves the overall performance of the model. Table 2 presents all the end-to-end architectures analyzed in this study. The selected architectures are the most popular and the most cited during the last eight years.

As is shown in Table 2, the first end-to-end architectures were based on LAS and RNN approaches. Then, different RNN approaches such as CNN, BiRNN, LSTM, GRU etc., were implemented. And today, the end-to-end ASR systems are focused on Transformers technology.

Referring to the WER results for each architecture analyzed (See Figure 22), end-to-end systems based on Transformers achieve state-of-the-art performance reaching WER values between 4-5%. Also, the end-to-end architectures based on CNN and BiRNN yield WER values between 4-5%. In addition, the implementation of language models reduces WER from 2-9% depending on the architecture where it is implemented. We noticed that the implementation of Spec Augment techniques reduces WER drastically.

5.1.3 ASR systems for low-resource language

Table 3 presents all architectures for low-resource languages analyzed in this study. The selected architectures are the most popular and the most cited for low-resource languages during the last five years. As is shown in Table 3, different hybrid architectures based on commercial HMM-GMM-DNN models and end-to-end models have been analyzed. Also, researchers have shown different perspectives related to end-to-end speech recognition systems. During the last two years, due to the extraordinary advantages of Transformers in the domain of speech recognition, researchers are mainly focused on the development of different architectures based on Transformers. One of the biggest difficulties in this session is the comparison of architectures to each other since each architecture is

trained and evaluated with a specific low-resource language. Each language has its characteristics, including morphological, syntactic, grammatical or semantic specifics. Also, dialects, socio-linguistic specifics, ethnocultural specifics or even the physical specifics of the person who articulates the word make this process more complex. In addition, the size, accuracy, speaker attributes etc., of the corpus are not the same, which directly affects the performance of the model.

Despite these limitations, Figure 34 clearly shows that architectures based on Transformers achieve state-of-the-art performance for low-resource languages. But even the end-to-end models based on CNN, BirNN approaches etc., by implementing Spec Augment techniques and different language models yield great results. It is noted that in Transformers-based architectures, the implementation of wav2vec approaches outperforms other Transformers-based architectures. In addition, the implementation of the language model reduces the WER by 2-6%

5.1.4 Transformers-based ASR systems

Transformers is a powerful deep learning model which is composed of two parts: an encoder and a decoder. It works through sequence-to-sequence learning where the Transformers take a sequence of tokens and predicts the next word in the output sequence. Unlike hybrid systems and end-to-end systems, Transformers uses an attention mechanism which provides context around items in the input sequence. So rather than start to run the first word of the sentence, the Transformers attempt to identify the context that brings meaning to each word of the sequence. This mechanism gives the Transformers a huge leg up over hybrid and end-to-end systems that must run in sequence, by improving the overall performance of the model. Transformers run multiple sequences in parallel and this accelerates training times greatly.

Table 4 presents all Transformers-based architectures analyzed in this study. The selected architectures are the most popular and the most cited during the last four years. All models analyzed are built over the baseline-Transformers architecture and are integrated with various approaches such as LAS, Augmentation, CTC, wav2vec etc., as shown in detail in Table 4. Most of the architectures analyzed were trained and evaluated with the LibriSpeech corpus, but not all. Despite this, the results presented in Figure 51 clearly showed the performance of each architecture. We note that all Transformers-based

architectures yield great WER results within the range of 3.3-12.9%. The best architecture implements the wav2vec approach and yields a WER of 3.3%. Also, we note that the application of language models reduces WER by 2 - 5% in some architectures.

5.2 Primary findings

Primary findings are based on objectives and hypotheses raised in our study. They derived directly from experiments conducted. Our experiments addressed the questions of whether 1) deep learning techniques are suitable for speech recognition of Albanian language; 2) the corpus specifications such as size, voice, dialect etc., affect the accuracy of the model; and 3) the end-to-end speech recognition architecture achieves state-of-the-art performance.

First, we investigated various architectures based on end-to-end approaches to identify the best architecture for the Albanian language. In addition, we evaluated the effect of corpus size, voice and dialect on the accuracy of the models as well as evaluated the CASR corpus in comparison to the LibriSpeech corpus.

Second, we investigated a Transformers-based architecture for the Albanian language and conducted a comparison of the proposed end-to-end architectures with the Transformers-based architecture.

5.2.1 Evaluation of end-to-end architectures for the Albanian language

During the evaluation of end-to-end architectures, we faced a high memory consumption, so it took us a long time to train our models. Although some solutions exist such as multi-time step parallelization, the proposed end-to-end architectures do not permit these types of parallelization. This issue is one of the biggest limitations faced by end-to-end models today. One of the biggest challenges in end-to-end models is to find the optimal model, a trade-off between the accuracy of the model and its complexity, to identify the best architecture. Models with a small number of layers generally are faced with over-fitting problems, while models with a large number of layers require a lot of training time, face hardware limitations and also have difficulties with unseen speech data. To find the best model, we evaluated five architectures with different numbers of layers. All architectures were configured with the same hyper-parameters (See Table 6) and run with the same hardware conditions. For each architecture, we

evaluated its accuracy by referring to standard parameters word error rate (WER) and character error rate (CER), as well as the training time. Due to the hardware limitations faced and the complex nature of the end-to-end models, the evaluation of these architectures is performed through the training set, which contains 100 hours of audio data along with their transcripts.

All experiments were performed for 100 epochs. The results show that the architecture which is composed of three RNN layers and five GRU layers outperformed the rest of the architectures yielding 5 % WER and 1 % CER. We noticed that with the increase in the number of layers, the performance of the model improves significantly. But with the increase in the number of layers, we faced a long time to train the models as well as a large hardware consumption (see Table 7). For models with more than 3 RNN and 5 GRU layers, we could not experiment since our hardware systems did not support the complexity of the model created. Referring to the experimental results at this phase, taking into account the accuracy of the model and its training time, in the trade-off between these two parameters, we conclude that the architecture with 1 RNN layer and 3 GRU layers is the best analyzed for SR of the Albanian language. Although the WER for this architecture is 2% more than the architecture with 3 RNN and 5 GRU layers and CER is 4% more, the training time of this architecture is reduced by 40%.

Following the objectives of our study, next, we evaluated the best architecture (1RNN and 3 GRU layers) through the testing set, to show the reaction for unknown data. To address this issue, we created a training set and a testing set with a split ratio of 80:20 according to academics, which means that 80 hours were used for training and 20 hours for testing. And referring to the experimental results, this architecture has yielded 11% CER and 35% WER on its testing set. By comparing the results for our model for both the training set and testing set, with the end-to-end models analyzed from the literature (See Table 2), we conclude that end-to-end deep learning techniques are suitable for Albanian ASR.

In this way, we answer the first research question that we have addressed in this study "Are deep learning techniques suitable for speech recognition of Albanian language?" and at the same time, we verified that the first hypothesis addressed "The deep learning techniques are suitable for SR of Albanian language" is true. So, the deep learning techniques have to consider a long design of speech recognition systems for the Albanian language.

5.2.2 Evaluation of CASR, and the effect of corpus size, voice and dialect on the accuracy of the model.

The development of ASR systems is based on corpus-driven techniques. Well-annotated speech corpus is the desirable quality of spoken language resources for the development and evaluation of ASR systems. So, a quality corpus should be checked manually with human intervention. To create a large quality corpus for research and development purposes, we need to give a great effort referring to speech data collection, annotation, validation, organization and documentation. In this study, we designed a corpus for the Albanian language called CASR. It contains 100 hours of speech recordings along with their transcripts and consists of a rich vocabulary that covers various topics such as biography, social and political sciences, psychology, religion, economics and business, history, philosophy and sociology, to be as heterogeneous as possible. During the design of the corpus, we considered the phonetics, semantics, morphology and syntax of the Albanian language. In addition, we considered the age, gender, accent, speed of utterance and dialect of speakers.

One of the limitations that we faced in this study was the lack of an existing corpus in the Albanian language to evaluate ASR systems, as well as a speech recognition system, to compare the results of our experiments in this study.

To address this issue, we evaluated the CASR in comparison to the LibriSpeech corpus. We created a training set and a testing set with a split ratio of 80:20 randomly for both corpora, where the training set was composed of 80 hours and the testing set was composed of 20 hours. In both cases, we choose the architecture with 1 RNN layer and 3 GRU layers, and the same hyper-parameters (See Table 6).

Referring to the experimental results our findings about WER are that model trained with CASR corpus yield a WER of 35% on its own test set, while, the model trained with LibriSpeech corpus yield a WER of 32% on its own test set.

While about CER, our findings are that models trained with CASR corpus yield a CER of 11% on their own test set, while, the model trained with the LibriSpeech corpus yields a CER of 9% on its own test set. Although there is a difference of 3% WER between CASR and LibriSpeech corpus, which is a negligible

value, we demonstrated that our corpus (CASR) has high accuracy, comparable to the LibriSpeech corpus which is an international standardized corpus for training and evaluated different ASR systems.

By the second research question addressed in this study and the second hypothesis, we evaluated the effect of corpus size, voice and dialect on the accuracy of the model.

First, we addressed the effect of corpus size on the accuracy of the model. For this purpose, we created four subsets with 1 hour, 10 hours, 50 hours and 100 hours (See Table 5), where each subset was divided into a training set and a testing set with a split ratio of 80:20 randomly. The evaluation was done on a testing set and the experiments were done with the same hyper-parameters in all cases, the architecture was used with 1 RNN layer and 3 GRU layers. Referring to experimental results, our finding is that the size of the corpus has a direct impact on the accuracy of the model. With the increase in corpus size, the WER and CER decrease linearly. The large corpora yield better accuracy and converge faster in this direction.

Second, we addressed the effect of the voice and dialect of the speaker on the accuracy of the model. For this purpose, we created two subsets of 20 hours, called Clean_CASR and Mix_CASR. Clean_CASR consisted of standard Albanian language without using dialects and included only one male speaker. Mix_CASR consisted of both the standard Albanian language and its dialects and included some speakers of both genders and different age groups from 20 to 70 years old. Even in this case, both subsets created were evaluated with the same architecture (1 RNN layer and 3 GRU layers) on its training set and with the same hyper-parameters (see Table 6). Referring to the experimental results our findings are that the dialect and voice of the speaker affect the accuracy of the model. For WER, the difference when the model was trained with the Clean_CASR corpus and the Mix_CASR corpus is 1%, while about CER the values are the same in both cases. Due to the limited resources, we had available regarding the size of the corpus to evaluate the effect of voice and dialect of the speaker on the accuracy of the model, we could not experiment more with these parameters. Although the difference is only 1%, it seems clear that these parameters affect the performance of the model.

In this way, our experiments addressed the answer to the second research question that we have raised in this study "How do corpus specifications (such as size, voice and dialect.) affect the accuracy of the model" at the same time we verified that the second hypothesis addressed "The corpus specifications

such as size, voice and dialect, affects the accuracy of the model" is true. So, these corpus parameters have to consider a long design of corpus.

5.2.3 Evaluation of Transformers-based architecture for the Albanian language.

Our experiments over the Transformers-based architecture addressed the third research question raised in this study "Which deep learning architectures for Albanian speech recognition achieve state-of-the-art performance?" For this purpose, we used the same training set and testing set, which trained the end-to-end model (1RNN and 3 GRU layers), so, the training set consisted of 80 hours and the testing set composed of 20 hours. The model was configured with hyper-parameters (see Table 8) and was evaluated on its testing set. Referring to the experimental results presented in Table 9, are impressive. Our finding is that Transformers-based architecture yields a WER of 18% on its testing set by achieving state-of-the-art performance for the Albanian language. In addition, Transformers-based architecture converges quickly, where up to epoch number 10, it reaches the minimum of WER. Also, an impressive result of the Transformers-based architecture is the low training time of the model. It only takes 72 hours to train. These experiments addressed the answer to the third research question that we have raised in this study and at the same time, we verified that the third hypothesis addressed "The end-to-end Deep SR model for the Albanian language achieve state-of-the-art WER (Word Error Rate) results" is not true. So, the Transformers-based architecture could be used as an opportunity to achieve state-of-the-art WER in SR systems for the Albanian language. Also, these results reinforce once again that the first hypothesis is true.

Based on all the experimental results in this study we find that the proposed Transformers-based architecture outperforms all end-to-end architectures proposed in this study in all directions. In Table 10, we briefly present a comparison between the end-to-end model and the Transformers-based model, by referring to our findings in this study.

Table 10. A comparison between end-to-end and Transformers-based models.

Findings	Transformers-based ASR system	End-to-end ASR system
WER % in the testing set	18%	35%
Training time	72 hours	722 hours
Convergence	10 epoch	100 epoch

5.3. Limitations

Our study has two main limitations. The first limitation is hardware resources. If the hardware resources are insufficient to train ASR models, they will multiply the training time of the model, and the complex models that have a large number of layers will be blocked during their training. The second limitation is the lack of previous research works on Albanian speech recognition. This limits us to compare our corpus (CASR) with other corpora in the Albanian language, as well as comparing the results of our architectures with previous architectures, to achieve state-of-the-art performance.

5.4 Conclusion

In this Chapter, we presented all the findings and discussions based on the results from the experimentations in Chapter 4 and Literature Review Chapter 2.

In Section 5.1, all the findings derived from the literature review called secondary findings have been shared. In this section, we identified the best speech recognition architectures during the last decade.

In Section 5.1, we have presented all findings derived from our experiments, called primary findings. In addition, we answered the research questions addressed in this thesis and discussed the truth of the hypotheses targeted.

Finally, in Section 5.4 all limitations of our research were presented.

6. CONCLUSIONS

In this thesis, we investigated speech recognition (SR) of Albanian language as a low-resource scenario of automatic speech recognition (ASR) system, by exploring various methods and architectures based on deep learning techniques.

In addition, we introduced a Corpus for Albanian Speech Recognition (CASR), aimed at training and evaluating various ASR systems. The corpus consists of 100 hours of audio recordings along with their transcripts. It contains a rich vocabulary which covers the topics such as biography, social and political sciences, psychology, religion, economics and business, history, philosophy and sociology, to be as heterogeneous as possible.

During the design of the corpus, we considered the phonetics, semantics, morphology and syntax of the Albanian language. In addition, we considered the age, gender, accent, speed of utterance and dialect of speakers, which are very important for a corpus.

The main goal of the thesis was to evaluate the three hypotheses addressed:

I. The deep learning techniques are suitable for the SR of the Albanian language.

II. The corpus specifications such as size, voice and dialect, affect the accuracy of the model.

III. The end-to-end Deep SR model for the Albanian language achieves state-of-the-art WER (Word Error Rate) results.

In addition, in this thesis we also targeted the three research questions:

I. Are deep learning techniques suitable for speech recognition of the Albanian language?

II. How do corpus specifications (such as size, voice and dialect.) affect the accuracy of the model?

III. Which deep learning architecture for speech recognition tasks performs best in the Albanian language?

Based on the experiments and results presented in Chapter 3, we accept the first and the second hypothesis, while we do not accept the third hypothesis.

The end-to-end ASR system for the Albanian language yields a WER of 7% and CER of 5% on its training set and a WER of 35% and CER of 11% on its testing set. The Transformers-based ASR system yields a WER of 18% on its testing set.

We conclude that the deep learning techniques have to consider a long design of speech recognition systems for the Albanian language. In addition, we conclude that Transformers-based architecture outperforms all end-to-end architectures proposed in this study by achieving state-of-the-art performance.

Also, we conclude that the corpus size, voice and dialect affect the accuracy of the model.

For the first research question, we analyzed the results of the experiments for both Albanian and English languages and by comparing them to each other, we concluded that the deep learning techniques are suitable for Albanian ASR systems.

For the second research question, we analyzed the results of the experiments with different corpus sizes (1, 10, 50 and 100 hours) as well as we analyzed two corpora called Clean_CASR and Mix_CASR. The Clean_CASR consisted of standard Albanian language and only one male speaker, while the Mix_CASR consisted of both the standard Albanian language and its dialects and included some speakers of both genders and different age groups from 20 to 70 years old. We conclude that the corpus size, voice and dialect affect the accuracy of the model.

For the third research question, we analyzed the experimental results of end-to-end models and Transformers-based models and concluded that Transformers-based architecture achieves state-of-the-art performance by reducing the WER by 17% on the testing set.

In chapter 1, we presented the importance of the thesis, including the hypothesis and research questions. We saw that this topic yields a noble contribution in the field of speech recognition for the Albanian language, which is expected to accelerate research within this domain.

In Chapter 2, we presented the results from the literature review process. We analyzed the most cited and popular ASR systems during the last decade and targeted Hybrid speech recognition systems, End-to-End speech recognition systems, ASR Systems for Low Resource Language and Transformers-based ASR systems. At the end of this chapter, we defined the best architectures, which helped our process for building the ASR systems for the Albanian language.

In Chapter 3, we presented the model design, corpus development, tools and assessment criteria according to the thesis objectives.

In Chapter 4, we presented all experimental results for the proposed models. We trained and evaluated the models with both the CASR and LibriSpeech corpora by the research questions and hypothesis addressed in this study.

Also, we evaluated the CASR corpus and its features, as well as we made a comparison with the LibriSpeech corpus.

In addition, we experimented with both training and testing subsets, to see the reaction of the models to known data and unknown data.

In Chapter 5, we presented all the findings and relevant discussion from our thesis. In Section 5.1, we presented all findings derived from the literature review called secondary findings. And, in section 5.2 we presented all findings derived from our experiments called primary findings. In addition, we answered the research questions targeted in this thesis and discussed the addressed hypothesis.

Finally, we faced two main limitations in this thesis. First, it was the hardware resources. This limited us to experiment with complex models which are built with more than 8 layers (3 RNN and 5 GRU), as well as with various hyper-parameters. We faced this limitation only for end-to-end models, as Transformers-based architectures exceeded this.

Second, there was no existing corpus in the Albanian language to train ASR systems, and there was no architecture designed for Albanian ASR. This limited us to compare our models with previous architectures, to achieve state-of-the-art performance.

This thesis presented a noble contribution to the field of natural language processing, with a focus on Albanian Speech Recognition, which is expected to accelerate research within the speech community for Albanian language and other low-resource languages.

Future research may extend this work by enhancing the proposed architectures and removing the limitations which we faced.

In addition, we are aiming at:

- To increase the size of the CASR corpus.
- To apply language models in the proposed architectures.
- To design other ASR architectures suitable for the Albanian language.
- To design a multilingual framework suitable for Albanian and other low-resource languages.

PUBLICATIONS AND PRESENTATIONS

Rista, A., & Kadriu, A. (2021). CASR: A Corpus for Albanian Speech Recognition. In 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 438-441). IEEE.

Rista, A., & Kadriu, A. (2021). End-to-End Speech Recognition Model Based on Deep Learning for Albanian. In 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 442-446). IEEE.

Rista, A., & Kadriu, A. (2020). Automatic Speech Recognition: A Comprehensive Survey. SEEU Review, 15(2), 86-112.

Rista, A., & Kadriu, A. (2022). A Model for Albanian Speech Recognition Using End-to-End Deep Learning Techniques. Interdisciplinary Journal of Research and Development, 9(3), 1-1.

REFERENCES

[1] Yu, D., & Deng, L. (2016). Automatic speech recognition (Vol. 1). Berlin: Springer.

[2] Juang, B. H., & Rabiner, L. R. (1991). Hidden Markov models for speech recognition. Technometrics, 33(3), 251-272.

[3] Reynolds, D. A. (2009). Gaussian mixture models. Encyclopedia of biometrics, 741(659-663).

[4] Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing, 20(1), 30-42.

[5] Graves, A., Jaitly, N., & Mohamed, A. R. (2013, December). Hybrid speech recognition with deep bidirectional LSTM. In 2013 IEEE workshop on automatic speech recognition and understanding (pp. 273-278). IEEE.

[6] Sun, R. H., & Chol, R. J. (2020). Subspace Gaussian mixture-based language modeling for large vocabulary continuous speech recognition. Speech Communication, 117, 21-27.

[7] Graves, A., & Jaitly, N. (2014, June). Towards end-to-end speech recognition with recurrent neural networks. In International conference on machine learning (pp. 1764-1772). PMLR.

[8] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C. ... & Zhu, Z. (2016, June). Deep speech 2: End-to-end speech recognition in English and mandarin. In International conference on machine learning (pp. 173-182). PMLR.

[9] Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016, March). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In the 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4960-4964). IEEE.

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

[11] Dong, L., Xu, S., & Xu, B. (2018, April). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5884-5888). IEEE.

[12] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in INTERSPEECH, Shanghai, China, Oct. 2020, pp.5036–5040.

[13] Mandic, D., & Chambers, J. (2001). Recurrent neural networks for prediction: learning algorithms, architectures and stability. Wiley.

[14] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84). Cham: Springer. U., Liu, J., & Whitaker, J. (2019). (Vol. 84). Cham: Springer.

[15] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. ArXiv preprint arXiv: 1511.08458.

[16] Vydana, H. K., & Vuppala, A. K. (2017). Residual neural networks for speech recognition. In 2017 25th European Signal Processing

Conference (EUSIPCO) (pp. 543-547). IEEE.

[17] Latif, S., Qadir, J., Qayyum, A., Usama, M., & Younis, S. (2020). Speech technology for healthcare: Opportunities, challenges, and state of the art. IEEE Reviews in Biomedical Engineering, 14, 342-356.

[18] Dalim, C. S. C., Sunar, M. S., Dey, A., & Billinghurst, M. (2020). Using augmented reality with speech input for non-native children's language learning. International Journal of Human-Computer Studies, 134, 44-64.

[19] Vajpai, J., & Bora, A. (2016). Industrial applications of automatic speech recognition systems. International Journal of Engineering Research and Applications, 6(3), 88-95.

[20] Abdulkareem, A., Somefun, T. E., Chinedum, O. K., & Agbetuyi, F. (2021). Design and implementation of speech recognition system integrated with internet of things. International Journal of Electrical and Computer Engineering (IJECE), 11(2), 1796-1803.

[21] Helmke, H., Kleinert, M., Ohneiser, O., Ehr, H., & Shetty, S. (2020, October). Machine learning of air traffic controller command extraction models for speech recognition applications. In 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC) (pp. 1-9). IEEE.

[22] Eddy, S. R. (1996). Hidden Markov models. Current opinion in structural biology, 6(3), 361-365.

[23] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E. ... & Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. ArXiv preprint arXiv: 1412.5567.

[24] Watanabe, S., Hori, T., Kim, S., Hershey, J. R., & Hayashi, T. (2017). Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing, 11(8), 1240-1253.

[25] Miao, Y., Gowayyed, M., & Metze, F. (2015, December). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 167-174). IEEE.

[26] Palaz, D., & Collobert, R. (2015). Analysis of CNN-based speech recognition system using raw speech as input (No. REP_WORK). Idiap.

[27] Li, J., Yu, D., Huang, J. T., & Gong, Y. (2012, December). Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM. In the 2012 IEEE Spoken Language Technology Workshop (SLT) (pp. 131-136). IEEE.

[28] Hossan, M. A., Memon, S., & Gregory, M. A. (2010, December). A novel approach for MFCC feature extraction. In 2010 4th International Conference on Signal Processing and Communication Systems (pp. 1-5). IEEE.

[29] Jaitly, N., Nguyen, P., Senior, A., & Vanhoucke, V. (2012). Application of pre-trained deep neural networks to large vocabulary speech recognition.

[30] Fischer, A., & Igel, C. (2012, September). An introduction to restricted Boltzmann machines. In Iberoamerican congress on pattern recognition (pp. 14-36). Springer, Berlin, Heidelberg.

[31] Kadyan, V., & Kaur, M. (2020). Sgmm-based modeling classifier for Punjabi automatic speech recognition system. In Smart Computing Paradigms: New Progresses and Challenges (pp. 149-155). Springer, Singapore.

[32] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. ... & Vesely, K. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society.

[33] Babich, G. A., & Camps, O. I. (1996). Weighted Parzen windows for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(5), 567-570.

[34] Passricha, V., & Aggarwal, R. K. (2020). A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. Journal of Intelligent Systems, 29(1), 1261-1274.

[35] Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A. ... & Ney, H. (2019). RWTH ASR Systems for LibriSpeech: Hybrid vs Attention--w/o Data Augmentation. ArXiv preprint arXiv: 1905.03072.

[36] R. Schl¨uter, I. Bezrukov, H. Wagner, and H. Ney, “Gamma-tone features and feature combination for large vocabulary speech recognition,” in Proc. ICASSP, Honolulu, HI, USA, Apr. 2007.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint ar X iv: 1412.6980, 2014.

[38] T. Dozat, “Incorporating nesterov momentum into Adam,” Stanford University, Tech. Rep., 2015. [Online]. Available: http://cs229.stanford.edu/proj2015/054 report.pdf.

[39] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schl¨uter, and H. Ney, “A comprehensive study of deep bidirectional lstm rnns fomodellingic modeling in speech recognition,” in Proc. ICASSP, New Orleans, LA, USA, Mar. 2017.

[40] R. Kneser and H. Ney, “Improved backing-off language modellingge modeling,” in Proc. ICASSP, Detroit, MI, USA, May 1995.

[41] M. Sundermeyer, R. Schl¨uter, and H. Ney, “LSTM neural networks for language modelling,” in Proc. Interspeech, Portland, OR, USA, Sep. 2012.

[42] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in ACL, Berlin, Germany, August 2016.

[43] Pan, J., Shapiro, J., Wohlwend, J., Han, K. J., Lei, T., & Ma, T. (2020). ASAPP-ASR: Multistream CNN and self-attentive SRU for SOTA speech recognition. ArXiv preprint arXiv: 2005.10469.

[44] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, “Semi-orthogonal low-rank matrix factorization for deep neural networks,” in Interspeech, 2018, pp. 3743–3747.

[45] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, “Simple recurrent units for highly parallelizable recurrence,” in EMNLP, 2018.

[46] Zeineldeen, M., Xu, J., Lüscher, C., Michel, W., Gerstenberger, A., Schlüter, R., & Ney, H. (2022, May). Conformer-based hybrid ASR system for Switchboard dataset. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7437-7441). IEEE.

[47] Tuske, P. Golik, R. Schluter, and H. Ney, “Speaker Adaptive Joint Training of Gaussian Mixture Models and Bottleneck Features,” in ASRU, Scottsdale, USA, Dec. 2015, pp.596–603.

[48] A. Krogh and J. Hertz, “A Simple Weight Decay Can Improve Generalization,” in NIPS, Colorado, USA, Dec. 1991.

[49] T. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, “Focal Loss for Dense Object Detection,” in ICCV, Venice, Italy, Oct.2017, pp. 2999–3007.

[50] K. Vesel`y, A. Ghoshal, L. Burget, and D. Povey, “Sequence discriminative Training of Deep Neural Networks,” in INTER- SPEECH, Lyon, France, Aug. 2013, pp. 2345–2349.

[51] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. IEEE transactions on acoustics, speech, and signal processing, 37(3), 328-339.

[52] Zhang, F., Wang, Y., Zhang, X., Liu, C., Saraf, Y., & Zweig, G. (2020). Faster, simpler and more accurate hybrid ASR systems using word pieces. ArXiv preprint arXiv: 2005.09150.

[53] O. Abdel-Hamid, A. Mohamed, H. Jiang et al., “Convolutional neural networks for speech recognition,” IEEE TASLP, 2014.

[54] A. Tjandra, C. Liu, F. Zhang et al., “Deja-vu: Double Feature Presentation and Iterated loss in Deep Transformer Networks,” in Proc. ICASSP, 2020.

[55] Y. Wang, A. Mohamed, D. Le, C. Liu et al., “Transformer-based Acoustic Modeling for Hybrid Speech Recognition,” in Proc. ICASSP, 2019.

[56] Y. Wu, M. Schuster, Z. Chen, Q. V. Le et al., “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” arXiv: 1609.08144, 2016.

[57] T. Kudo, “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates,” in Proc. ACL, 2018.

[58] Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J. ... & Seltzer, M. L. (2020, May). Transformer-based acoustic modelling for hybrid speech recognition. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6874-6878). IEEE.

[59] A. Radford, K. Narasimhan, Tim S., et al., “Improving language understanding by generative pre-training,” 2018.

[60] O. Myle, E. Sergey, B. Alexei, F. Angela, et al., “fairseq: A Fast, Extensible Toolkit for Sequence Modeling,” in Proceedings of NAACL- HLT 2019: Demonstrations, 2019.

[61] Sainath, T. N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A. R., Dahl, G., & Ramabhadran, B. (2015). Deep convolutional neural networks for large-scale speech tasks. Neural networks, 64, 39-48.

[62] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). IEEE.

[63] Iba, H., & Noman, N. (2020). Deep Neural Evolution. Berlin: Springer.

[64] Song, W., & Cai, J. (2015). End-to-end deep neural network for automatic speech recognition. Standford CS224D Reports.

[65] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of momentum and initialization in deep learning. In 30th International Conference on Machine Learning, 2013.

[66] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, and B. Catanzaro. Deep learning with COTS HPC. In International Conference on Machine Learning, 2013.

[67] Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. abs/1502.03167, 2015. http://arxiv.org/abs/1502.03167.

[68] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-Based Models for Speech Recognition. In http://arxiv.org/abs/1506.07503, 2015.

[69] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. (2016, March). End-to-end attention-based large vocabulary speech recognition. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4945-4949). IEEE.

[70] Zhang, Y., Chan, W., & Jaitly, N. (2017, March). Very deep convolutional networks for end-to-end speech recognition. In 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4845-4849). IEEE.

[71] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in NIPS, 2015.

[72] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., “Streaming end-to-end speech recognition for mobile devices,” in Proc. ICASSP.

[73] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. Chiu, and A. Kannan, “Minimum Word Error Rate Training for Attention-based Sequence-to-sequence Models,” in Proc. ICASSP (accepted), 2018.

[74] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks,” in Proc. NIPS, 2015, pp. 1171–1179.

[75] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[76] P. Goyal, P. Doll´ar, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: training imagenet in 1 hour,” CoRR, vol. abs/1706.02677, 2017.

[77] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” arXiv preprint arXiv:1608.06993, 2016

[78] J. Tang, Y. Song, L. Dai, and I. McLoughlimodellingstic modeling with the densely connected residual network for multichannel speech recognition,” in Proc. Interspeech 2018, 2018, pp. 1783–1787.

[79] Park, D. S., Chan, W., Zhang, Y., Chiu, C. C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779.

[80] William Chan, Yu Zhang, Quoc Le, and Navdeep Jaitly, “Latent Sequence Decompositions,” in ICLR, 2017.

[81] 2019, pp. 6381–6385. Li, J., Zhao, R., Hu, H., & Gong, Y. (2019, December). Improving RNN transducer modelling for end-to-end speech recognition. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 114-121). IEEE.

[82] Zhang, Q., Lu, H., Sak, H., Tripathi, A., McDermott, E., Koo, S., & Kumar, S. (2020, May). Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7829-7833). IEEE.

[83] Zihang Dai, Zhilin Yang, Yiming Yang, William W Co-hen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdi-nov, “Transformer-xl: Attentive language models beyond a fixed-length context,” in Proceedings of the 57th Annual Meet- ing of the Association for Computational Linguistics, 2019, p.29782988.

[84] Synnaeve, G., Xu, Q., Kahn, J., Likhomanenko, T., Grave, E., Pratap, V. ... & Collobert, R. (2019). End-to-end asr: from supervised to semi-supervised learning with modern architectures. ArXiv preprint arXiv:1911.08460.

[85] Hannun, A., Lee, A., Xu, Q., and Collobert, R. Sequence-to-sequence speech recognition with time-depth separable convolutions. Interspeech 2019, Sep 2019. DOI: 10.21437/interspeech.2019-2460.

[86] Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modelling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, pp. 933–941.JMLR.org, 2017.

[87] Baevski, A. and Auli, M. Adaptive input representations for neural language modelling. In International Conference on LearningRepresentations, 2019. URL, https://openreview.net/forum?id=ByxZX20qFQ.

[88] Likhomanenko, T., Synnaeve, G., and Collobert, R. Who needs words? lexicon-free speech recognition. ArXivpreprint arXiv: 1904.04479, 2019.

[89] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Jul. 2019, pp. 1810–1822.

[90] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," arXiv preprint ar X iv: 1906.02762, 2019.

[91] Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V. ... & Zhang, Y. (2020, May). Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6124-6128). IEEE.

[92] Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J.M. Cohen, H. Nguyen, and R.T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model,"arXiv:1904.03288, 2019.

[93] Vegesna, V. V. R., Gurugubelli, K., Vydana, H. K., Pulugandla, B., Shrivastava, M., & Vuppala, A. K. (2017, December). DNN-HMM acoustic modelling for large vocabulary telugu speech recognition. In International Conference on Mining Intelligence and Knowledge Exploration (pp. 189-197). Springer, Cham.

[94] Saul, L.K., Rahim, M.G.: Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. IEEE Trans. Speech Audio Process. 8(2), 115–125 (2000).

[95] Levinson, S. E., Rabiner, L. R., & Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell System Technical Journal, 62(4), 1035-1074.

[96] Digalakis, V., Rohlicek, J. R., & Ostendorf, M. (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. IEEE Transactions on speech and audio processing, 1(4), 431-442.

[97] Fathima, N., Patel, T., Mahima, C., & Iyengar, A. (2018, September). TDNN-based Multilingual Speech Recognition System for Low Resource Indian Languages. In Interspeech (pp. 3197-3201).

[98] Zhou, S., Xu, S., & Xu, B. (2018). Multilingual end-to-end speech recognition with a single transformer on low-resource languages. ArXiv preprint arXiv: 1806.05059.

[99] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” arXiv preprint ar X iv: 1508.07909, 2015.

[100] Baevski, A., Schneider, S., & Auli, M. (2019). Vq-wav2vec: Self-supervised learning of discrete speech representations. ArXiv preprint arXiv: 1910.05453.

[101] Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W Black, et al. The zero resource speech challenge 2019: Tts without t. arXiv, 1904.11469, and 2019.

[102] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. CoRR, abs/1904.05862, 2019. URL http://arxiv.org/abs/1904.05862.

[103] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. ArXiv, abs/1611.01144, 2016.

[104] Ryan Eloff, Andr´e Nortje, Benjamin van Niekerk, Avashna Govender, Leanne Nortje, Arnu Pretorius, Elan Van Biljon, Ewald van der Westhuizen, Lisa van Staden, and Herman Kamper. Unsupervised acoustic unit discovery for speech synthesis using discrete laten variable neural networks. arXiv, abs/1904.07556, 2019.

[105] A¨aron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv, abs/1807.03748, 2018

[106] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv, abs/1810.04805, 2018.

[107] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized Bert pretraining approach. ArXiv preprint arXiv: 1907.11692, 2019.

[108] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2letter: an end-to-end content-based speech recognition system. ArXiv, abs/1609.03193, 2016.

[109] Chen, Z., & Yang, H. (2020, June). Yi language speech recognition using deep learning methods. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 1064-1068). IEEE.

[110] Wang, W., Yang, X., & Yang, H. (2020, September). End-to-End low-resource speech recognition with a deep CNN-LSTM encoder. In 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP) (pp. 158-162). IEEE.

[111] Anoop, C. S., & Ramakrishnan, A. G. (2021, July). CTC-based end-to-end ASR for the low resource Sanskrit language with spectrogram augmentation. In 2021 National Conference on Communications (NCC) (pp. 1-6). IEEE.

[112] M. Nguyen, “Building an end-to-end speech recognition model in PyTorch,” https://www.assemblyai.com/blog/end-to-end-speech-recognition-pytorch, 2021.

[113] T. Zenkel, R. Sanabria, F. Metze, J. Niehues, M. Sperber, S. Stüker, and A. Waibel, “Comparison of decoding strategies for CTC acoustic models,” in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2017, vol. August, pp. 513–517.

[114] Zhang, Z. (2018, June). Improved adam optimizer for deep neural networks. In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS) (pp. 1-2). Ieee.

[115] M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” Computer, Speech & Language, vol. 16, no. 1, pp. 69–88, 2002.

[116] Meng, L., Xu, J., Tan, X., Wang, J., Qin, T., & Xu, B. (2021, June). MixSpeech: Data augmentation for low-resource automatic speech recognition. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7008-7012). IEEE.

[117] Xue, B., Yu, J., Xu, J., Liu, S., Hu, S., Ye, Z. ... & Meng, H. (2021, June). Bayesian transformer language models for speech recognition. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7378-7382). IEEE.

[118] Ke Li, Zhe Liu, Tianxing He, Hongzhao Huang, and Sanjeev Khudanpur, “An empirical study of transformer-based neural language model adaptation,” in ICASSP, 2020.

[119] Dan Hendrycks and Kevin Gimpel, “Bridging nonlinearities and stochastic regularizers with gaussian error linear units,” CoRR, vol. abs/1606.08415, 2016.

[120] Diederik P Kingma and Max Welling, “Auto-encoding variational Bayes,” arXiv preprint ar X iv: 1312.6114, 2013.

[121] Baevski, A., Hsu, W. N., Conneau, A., & Auli, M. (2021). Unsupervised speech recognition. Advances in Neural Information Processing Systems, 34, 27826-27839.

[122] Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with Gumbel-softmax. ArXiv preprint arXiv: 1611.01144.

[123] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. Proc. of NIPS, 2014.

[124] Guo, P., Boyer, F., Chang, X., Hayashi, T., Higuchi, Y., Inaguma, H. ... & Zhang, Y. (2021, June). Recent developments on espnet toolkit boosted by conformer. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5874-5878). IEEE.

[125] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, et al., "Conformer: Convolution-augmented transformer for speech recognition," arXiv preprint ar X iv: 2005.08100, 2020.

[126] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, et al., "Understanding and improving Transformer from a multi-particle dynamic system point of view," in Proc. ICLR, 2020.

[127] Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, et al., "Improving Transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in Proc. INTERSPEECH, 2019, pp. 1408–1412.

[128] Thienpondt, J., & Demuynck, K. (2022). Transfer Learning for Robust Low-Resource Children's Speech ASR with Transformers and Source-Filter Warping. ArXiv preprint arXiv: 2206.09396.

[129] G. Fant, "The source filter concept in voice production," STL-QPSR, vol. 1, no. 1981, pp. 21–37, 1981.

[130] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (step) improve speech recognition," in Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, vol. 117, 2013, p. 21.

[131] Bao Thai ; Robert Jimerson ; Dominic Arcoraci ; Emily Prud'hommeaux ; Raymond Ptucha"SYNTHETIC DATA AUGMENTATION FOR IMPROVING LOW-RESOURCE ASR", Western New York Image and Signal Processing Workshop (WNYISPW), IEEE 2019.

[132] Kenneth Heafield, "Kenlm: Faster and smaller language model queries," in Proceedings of the sixth workshop on statistical machine translation. Association for Computational Linguistics, 2011.

[133] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., “Deep speech: Scaling up end-to-end speech recognition,” arXiv preprint arXiv:1412.5567, 2014.

[134] Alex Graves, Santiago Fernandez, Faustino Gomez, and ´ Jurgen Schmidhuber, “Connectionist temporal classifi- ¨ cation: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[135] Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, “Letter-based speech recognition with gated convnets,” CoRR, vol. abs/1712.09444, 2017.

[136] Robbie Jimerson, Kruthika Simha, Raymond Ptucha, and Emily Prudhommeaux, “Improving asr output for endangered language documentation,” in Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages, 2018

[137] Thai, B., Jimerson, R., Ptucha, R., & Prud’hommeaux, E. (2020, May). Fully Convolutional ASR for Less-Resourced Endangered Languages. In Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and

[138] Collaboration and Computing for Under-Resourced Languages (CCURL) (pp. 126-130).

[139] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino et al., “Xls-r: Self-supervised cross-lingual speech representation learning at scale,” arXiv preprint arXiv:2111.09296, 2021.

[140] D. Hendrycks and K. Gimpel, “Gaussian error linear units (genus),” arXiv preprint ar X iv: 1606.08415, 2016.

[141] S. Shahnawazuddin, N. Adiga, K. Kumar, A. Poddar, and W. Ahmad, “Voice Conversion Based Data Augmentation to Improve Children’s Speech Recognition in Limited Data Scenario,” in Proc. Interspeech 2020, 2020, pp. 4382–4386.

[142] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.

[143] Dong, L., Xu, S., & Xu, B. (2018, April). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5884-5888). IEEE.

[144] Hrinchuk, O., Popova, M., & Ginsburg, B. (2020, May). Correction of automatic speech recognition with transformer sequence-to-sequence model. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7074-7078). IEEE.

[145] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," Inter-speech, 2019.

[146] B. Ginsburg, P. Castonguay, O. Hrinchuk, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, H. Nguyen, and J. M. Cohen, "Stochastic gradient methods with layer-wise adaptive moments for the training of deep networks," arXiv preprint arXiv:1905.11286, 2019.

[147] K. Heafield, "Kenelm: Faster and smaller language model queries," in Proceedings of the sixth workshop on statistical machine translation. Association for Computational Linguistics, 2011, pp. 187–197

[148] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," arXiv preprint arXiv:1901.02860, 2019.

[149] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," NAACL, 2019.

[150] Moriya, T., Ochiai, T., Karita, S., Sato, H., Tanaka, T., Ashihara, T. ... & Delcroix, M. (2020, October). Self-Distillation for Improving CTC-Transformer-Based ASR Systems. In INTERSPEECH (pp. 546-550).

[151] S. Karita, N. Yalta, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in Proc. of INTERSPEECH, 2019, pp.1408–1412.

[152] T. Moriya, S. Ueno, Y. Shinohara, M. Delcroix, Y. Yamaguchi, and Y. Aono, “Multi-task learning with augmentation strategy for acoustic-to-word attention-based encoder-decoder speech recognition,” in Proc. of INTERSPEECH, 2018, pp. 2399–2403.

[153] Inaguma, H., Cho, J., Baskar, M. K., Kawahara, T., & Watanabe, S. (2019, May). Transfer learning of language-independent end-to-end ASR with language model fusion. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing.

[154] Wang, D., & Zheng, T. F. (2015, December). Transfer learning for speech and language processing. In 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA) (pp. 1225-1237). IEEE.

[155] (ICASSP) (pp. 6096-6100). IEEE.

[156] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, “Advances in joint CTC-Attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM,” in Proc. of INTERSPEECH, 2017, pp. 949–953.

[157] A. Baevski, M. Auli, and A. Mohamed. Effectiveness of self-supervised pre-training for speech recognition. ArXiv, abs/1911.03912, 2019.

[158] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-softmax. ArXiv, abs/1611.01144, 2016.

[159] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modelling. In Proc. of NAACL System Demonstrations, 2019.

[160] Dalmia, S., Liu, Y., Ronanki, S., & Kirchhoff, K. (2021, June). Transformer-transducers for code-switched speech recognition. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5859-5863). IEEE.

[161] Chen, N., Watanabe, S., Villalba, J., Żelasko, P., & Dehak, N. (2020). Non-autoregressive transformer for speech recognition. IEEE Signal Processing Letters, 28, 121-125.

[162] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” NAACL, 2019.

[163] Bu, H., Du, J., Na, X., Wu, B., & Zheng, H. (2017, November). Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In 2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA) (pp. 1-5). IEEE.

[164] Maekawa, K. (2003). Corpus of Spontaneous Japanese: Its design and evaluation. In ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition.

[165] Alexei Baevski, Steffen Schneider, and Michael Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in Proc. of ICLR, 2020.

[166] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in Proc. of NAACL, 2019.

[167] Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer, “Transformers with convolutional context for asr,” arXiv, 2019.

[168] Mohamed, A., Okhonko, D., & Zettlemoyer, L. (2019). Transformers with convolutional context for asr. ArXiv preprint arXiv: 1904.11660.

[169] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,”2012. [Online]. Available: http://arxiv.org/abs/1212.5701.

[170] Fan, R., Chu, W., Chang, P., & Xiao, J. (2021, June). Cass-nat: Ctc alignment-based single-step non-autoregressive transformer for speech recognition. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5889-5893). IEEE.

[171] Fan, R., Chu, W., Chang, P., Xiao, J., & Alwan, A. (2021). An Improved Single Step Non-autoregressive Transformer for Automatic Speech Recognition. ArXiv preprint arXiv: 2106.09885.

[172] B. Yang, L. Wang, D. F. Wong, L. S. Chao, and Z. Tu, “Convolutional self-attention networks,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Jun. 2019, pp. 4040–4045.

[173] Shi, Y., Wu, C., Wang, D., Xiao, A., Mahadeokar, J., Zhang, X. ... & Seltzer, M. (2022, May). Streaming Transformer Transducer Based Speech Recognition Using Non-Causal Convolution. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 8277-8281). IEEE.

[174] Ching Feng Yeh, Yongqiang Wang, Yangyang Shi, et al., “Streaming attention-based models with augmented memory for end-to-end speech recognition,” in Proc. SLT, 2020

[175] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, and Others, “Emformer: Efficient Memory Transformer Based Acoustic Model For Low Latency Streaming Speech Recognition,” in Proc. ICASSP, 2021.

[176] Wang, C., Wu, Y., Chen, S., Liu, S., Li, J., Qian, Y., & Yang, Z. (2021). Self-Supervised Learning for speech recognition with Intermediate layer supervision. ArXiv preprint arXiv: 2112.08778.

[177] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in NAACL-HLT 2019, Jill Burstein, Christy Doran, and Thamar Solorio, Eds. 2019, pp. 4171–4186, Association for Computational Linguistics.

[178] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” IEEE ACM Trans. Audio Speech Lang. Process., vol. 29, pp. 3451–3460, 2021.

[179] Deng, K., Yang, Z., Watanabe, S., Higuchi, Y., Cheng, G., & Zhang, P. (2022). Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models. ArXiv preprint arXiv: 2201.10103.

[180] F. Yu and K. Chen, “Non-autoregressive transformer-based end-to-end ASR using BERT,” arXiv preprint abs:2104.04805, 2021.

[181] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline,” in Proc. O-COCOSDA, 2017, pp. 1–5.

[182] J. J. Godfrey, E. C. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone speech corpus for research and development,” in Proc. ICASSP, 1992, pp. 517–520.

[183] Yang, Y., Wang, P., & Wang, D. (2022). A Conformer Based Acoustic Model for Robust Automatic Speech Recognition. ArXiv preprint arXiv: 2203.00725.

[184] Fu, P., Liu, D., & Yang, H. (2022). LAS-Transformer: An Enhanced Transformer Based on the Local Attention Mechanism for Speech Recognition. Information, 13(5), 250.

[185] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778

[186] Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y.; Wang, L.; Liu, T. On-layer normalization in the Transformer architecture. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 10524–10533.

[187] Karita, S.; Soplin, N.E.Y.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In Proceedings of the Interspeech 2019, ISCA, Graz, Austria, 15–19 September 2019; pp. 1408–1412.

[188] Paçarizi, R. (2008). Albanian Language

[189] Orel, V. Ė. (2000). A concise historical grammar of the Albanian language: reconstruction of Proto-Albanian. Brill.

[190] Kadriu, A. (2010). Modelling a two-level formalism for inflexion of nouns and verbs in Albanian. Modeling Simulation and Optimization-Focus on Applications.

[191] Opitz, A. (2006). A reanalysis of definiteness-markers in Albanian noun inflexion. A subanalysis of Argument Encoding in Distributed Morphology, 84, 103-114.

[192] Kadriu, A. (2010). Modelling a two-level formalism for inflexion of nouns and verbs in Albanian. Modeling Simulation and Optimization-Focus on Applications.

[193] Güçlü, R. (2015). Adverb formation process in Albanian and Bodo Languages: A comparative study. International Journal of Social Sciences and Education Research, 3(5 S), 1842-1850.

[194] Vydana, H. K., & Vuppala, A. K. (2017). Residual neural networks for speech recognition. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 543-547). IEEE.

[195] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84). Cham: Springer. U., Liu, J., & Whitaker, J. (2019). (Vol. 84). Cham: Springer.

[196] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In the 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[197] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning (pp. 369-376).

[198] Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. ArXiv preprint arXiv: 1612.02295.

[199] Battenberg, E., Chen, J., Child, R., Coates, A., Li, Y. G. Y., Liu, H. ... & Zhu, Z. (2017, December). Exploring neural transducers for end-to-end speech recognition. In 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 206-213). IEEE.

[200] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. ArXiv preprint arXiv: 1711.05101.

[201] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbour search. IEEE Trans. Pattern Anal. Mach. Intell., 33(1):117–128, Jan. 2011.

[202] D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recog-nition,” arXiv:1904.08779 [eess.AS], Apr. 18 2019.

[203] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv, abs/1810.04805, 2018.

[204] Jegou, H., Douze, M., & Schmid, C. (2010). Product quantization for nearest neighbour search. IEEE transactions on pattern analysis and machine intelligence, 33(1), 117-128.

[205] Audacity, T. (2017). Audacity. The Name Audacity (R) Is a Registered Trademark of Dominic Mazzoni Retrieved from http://audacity. sourceforge. net.

[206] Fiscus, J. G., Ajot, J., Radde, N., & Laprun, C. (2006, May). Multiple dimension levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06).

[207] Morris, A. C., Maier, V., & Green, P. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In Eighth International Conference on Spoken Language Processing.

[208] Chung, J. S., Senior, A., Vinyals, O., & Z isserman, A. (2017, July). Lip reading sentences in the wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3444-3453

PhD thesis

A Framework for Albanian Speech Recognition using Deep Learning Techniques

Proofreading

I, Edmond Fejzulla, translator/interpreter-proof-reader from English into Albanian and vice-versa, certify that the PhD thesis "A Framework for Albanian Speech Recognition using Deep Learning Techniques" has been proofread by me and most shortcomings have been eliminated. The paper in question meets the languages standards to be defended as a PhD thesis.