ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

AMHARIC SPEECH RECOGNITION SYSTEM USING JOINT TRANSFORMER AND CONNECTIONIST TEMPORAL CLASSIFICATION WITH EXTERNAL LANGUAGE MODEL INTEGRATION

BY
ALEMAYEHU YILMA

ADVISOR
DR. BISRAT DEREBSSA

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering

JUNE, 2023
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

The undersigned have examined the thesis titled:

AMHARIC SPEECH RECOGNITION SYSTEM USING JOINT TRANSFORMER AND CONNECTIONIST TEMPORAL CLASSIFICATION WITH EXTERNAL LANGUAGE MODEL INTEGRATION

BY

ALEMAYEHU YILMA

Approval by Boards of Examiners

Dr. Bisrat Derebssa	_____	_____
Dean, SECE, AAiT	Date	Signature
Dr. Bisrat Derebssa	_____	_____
Advisor	Date	Signature
Dr. Fitsum Assamnew	_____	_____
Internal Examiner	Date	Signature
Dr. Menore Tekeba	_____	_____
External Examiner	Date	Signature

Declaration

I, Alemayehu Yilma Demisse, thus certify that this thesis is entirely my own work. All sources of information used in this study were properly cited and acknowledged. I further affirm that this thesis has not been submitted in part or in full to any other learning institution for any other requirements.

Declared By:

_____

Student's Name and Signature

JUNE, 2023

Acknowledgments

Prior to anything else, I would like to thank God in the most sincere way possible for leading me through every phase of my research journey.

I would also like to express my gratitude to Dr. Bisrat Derebssa, who served as my thesis advisor, for his constant support and invaluable advice during the whole process. He provided timely suggestions, helpful criticism, and comments that helped me refine my research and analytical abilities.

My sincere gratitude also extends to my family for their unfailing love, inspiration, and support. I will always be indebted to them for their unceasing encouragement and faith in me, which kept me going even through the most trying circumstances.

Last but not least, I extend my deepest appreciation to everyone who contributed to the success of my research project, whether by providing me with the necessary resources or providing insightful comments and suggestions. Your contributions have been invaluable, and I am profoundly grateful for your time, effort, and willingness to assist me in my academic pursuit.

Thank you all for your support and kindness.

Abstract

Sequence-to-sequence (S2S) attention-based models are deep neural network models that have demonstrated some tremendously remarkable outcomes in automatic speech recognition (ASR) research. In these models, the cutting-edge Transformer architecture has been extensively employed to solve a variety of S2S transformation problems, such as machine translation and ASR. This architecture does not use sequential computation, which makes it different from recurrent neural networks (RNNs) and gives it the benefit of a rapid iteration rate during the training phase. However, according to the literature, the overall training speed (convergence) of Transformer is relatively slower than RNN-based ASR. Thus, to accelerate the convergence of the Transformer model, this research proposes joint Transformer and connectionist temporal classification (CTC) for Amharic speech recognition system. The research also investigates an appropriate recognition units: characters, subwords, and syllables for Amharic end-to-end speech recognition systems. In this study, the accuracy of character- and subword-based end-to-end speech recognition system is compared and contrasted for the target language. For the character-based model with character-level language model (LM), a best character error rate of 8.84% is reported, and for the subword-based model with subword-level LM, a best word error rate of 24.61% is reported. Furthermore, the syllable-based end-to-end model achieves a 7.05% phoneme error rate and a 13.3% syllable error rate without integrating any language models (LMs).

Keywords: CTC, ASR, LM, S2S, RNN, Transformer

Table of Contents

Declaration i

Acknowledgments ii

Abstract iii

List of Figures vii

List of Tables viii

List of Acronyms ix

Chapter 1 **1**

1 Introduction **1**

1.1 Background 1

1.2 Amharic Language 3

1.2.1 Amharic Phonetics 4

1.2.2 Amharic Morphology 6

1.2.3 Amharic Syllabification 7

1.2.4 Amharic Writing System 8

1.3 Problem Statement 9

1.4 Objectives 10

1.4.1 General Objective 10

1.4.2 Specific Objectives 10

1.5 Contribution 11

1.6 Scope and Limitation 11

1.7 Document Organization 12

Chapter 2 **13**

2 Literature Review **13**

Chapter 3 **18**

3 Methodology **18**

3.1 Dataset 18

3.2 Proposed Model Architecture 20

3.2.1 Feature Extraction 21

3.2.1.1 Sampling and Quantization 21

3.2.1.2 Windowing 21

3.2.1.3 Discrete Fourier Transform 23

3.2.1.4 Mel filterbank and log 24

3.2.2 Sub-sampling 25

3.2.3 Acoustic Modeling 26

3.2.3.1 Transformer Architecture 27

3.2.3.2 Connectionist Temporal Classification (CTC) 29

3.2.3.3 Joint Transformer and CTC 30

3.2.4 Language Modeling 31

3.2.5 Joint Decoding 32

3.3 Decoding techniques 32

3.3.1 Greedy search 33

3.3.2 Beam search 34

3.4 Fundamental Amharic speech recognition units 35

3.5 ASR evaluation metrics 37

3.6 Summary 38

Chapter 4 **39**

4 Result and Discussion **39**

4.1 Experiment Setup 39

4.2 Experiment Results 40

4.2.1 Investigation of joint Transformer and Connectionist Temporal Classification (CTC) model 40

4.2.2 Investigation of Language Model 42

4.2.3 Impact of joint decoding 43

4.2.3.1 Character-based acoustic model 43

4.2.3.2 Subword-based acoustic model 44

4.2.3.3 Syllable-based acoustic model 45

4.2.4 Training and inference time 46

4.3 Discussion 47

Chapter 5 **50**

5 Conclusion and Future Work **50**

5.1 Conclusion 50

5.2 Future Work 51

Bibliography 52

List of Figures

3.1 The proposed model architecture for Amharic speech recognition system . 20

3.2 Windowing with a 10ms stride and a 25 ms rectangular window 22

3.3 Windowing a sine wave with the rectangular and Hamming windows . . . 23

3.4 (a) A 25 ms Hamming-windowed portion of the vowel 'e' and (b) its spectrum computed by a DFT 24

3.5 The mel triangular filter bank 25

3.6 Schematic architecture that shows pre-encoder stages 26

3.7 Greedy search algorithm 33

3.8 Beam search algorithm 34

4.1 Training losses in character-based Acoustic Model (AM) 41

4.2 Training and Validation perplexities of character-level Language Model (LM) 43

4.3 Training and Validation perplexities of subword-level LM 43

4.4 Training and inference time of Transformer-CTC and RNN-CTC 47

List of Tables

1.1 Categories of Amharic Consonants 4

1.2 Categories of Amharic Vowels 6

4.1 Decoding results of character based acoustic model 43

4.2 Decoding results of subword based acoustic model 45

4.3 Decoding results of syllable based acoustic model 46

List of Acronyms

AM Acoustic Model

ASR Automatic Speech Recognition

ASRS Automatic Speech Recognition System

BPE Byte Pair Encoding

CER Character Error Rate

CNN Convolutional Neural Network

CTC Connectionist Temporal Classification

CV Consonant-Vowel

DFT Discrete Fourier Transform

DNN Deep Neural Network

FF Feed Forward

GMM Gausian Mixture Model

HMM Hidden Markov Model

HTK Hidden Toolkit

IPA International Phonetic Alphabet

LM Language Model

LPCM Linear Pulse Code Modulation

LSTM Long Short Term Memory

LVCSR Large Vocabulary Continous Speech Recognition

MHA Multi Head Attention

OOV Out Of Vocabulary

PE Positional Encoding

PER Phoneme Error Rate

RNN Recurrent Neural Network

RNNLM Recurrent Neural Network Language Model

SER Syllable Error Rate

WER Word Error Rate

Chapter 1

Introduction

1.1 Background

Automatic Speech Recognition (ASR) has a wide range of applications in security, education, e-health, and transport systems, making it an important and active research domain. ASR is the process by which the information conveyed by a speech signal is decoded and transcribed into a set of characters, or graphemes.

Research in ASR has been sparked by improvements in computer technology, notably in terms of storage capacity, processing speed, and other speech processing requirements. As a result, a lot of research has been conducted on the development of ASR systems, which has led to the development of many ASR systems. So far, however, advanced speech recognition systems have only been developed for technologically favorable languages such as English, European, and Asian languages [1]. Research on developing ASR systems for technologically unfavorable languages like Amharic is still in its early stages. The lack of a large corpus or dataset of standard Amharic language speech is to blame for this.

The traditional ASR systems consist of three independent components: an Acoustic Model (AM), a Language Model (LM) and a lexicon [2]. The LM was developed using n-gram models, whereas the AM was predicated on a Hidden Markov Model (HMM) and Gausian Mixture Model (GMM) (HMM-GMM) framework. These conventional systems were difficult to manage and configure since each component was trained separately. This reduced the efficacy of using these systems. With the advent of deep learning, ASR systems have become more successful. As opposed to GMM, Deep Neural Network (DNN) started to be employed for acoustic modeling, which produced better results in numerous research papers [1, 3, 4, 5]. As a result, one of the most commonly utilized models for continuous speech recognition was the hybrid HMM-DNN architecture.

In recent years, ASR systems have undergone a discernible transition from a hybrid HMM - DNN modeling approach [6] to an end-to-end or all neural networks modeling approach [7, 8]. In contrast to the traditional model, which comprises a number of independent components, the end-to-end structure portrays the system as a single neural network [9]. It is simple to develop end-to-end systems because they instantly transform an input sequence of acoustic signals into an output label sequence without the need for intermediary states or the requirement for further processing at the output. The major challenges associated with the availability of a high-performance computing environment and the gathering of a significantly large number of speech data must be accomplished in order to improve the performance of end-to-end systems. The successful application of deep learning systems, including speech recognition systems, depends on these issues being solved.

End-to-end systems are exemplified by models like the CTC [10] and the attention-based encoder-decoder [11]. The CTC-based acoustic model training doesn't need frame-level alignments between characters in the transcript and the observed input speech. This is due to CTC introducing a "blank label," which determines the start and end of one character, [10]. In the attention-based encoder-decoder models, the encoder is analogous to AM that transforms input speech into higher-level representation, the decoder is analogous to LM that predicts each output token as a function of the prior prediction, and the attention mechanism is an alignment model that determines frames in the encoder representation that the decoder should attend to in order to predict the next token [12].

Recurrent Neural Network (RNN) is the basis of the aforementioned end-to-end models. RNN-based models produce a sequence of hidden layers based on the network's prior hidden layer by performing computations on the character positions of the received and resulting data. Because this sequential procedure prevents parallel computation, training the model with a longer input sequence takes much more time. In order to reduce sequential processes, the model architecture of the Transformer has been proposed in [13]. This architecture eliminates recurrence and relies on its internal attention (self-attention) mechanism without using RNN to determine dependencies between input and output data, which allows parallelization of the training process. The fast rate of learning due to the absence of sequential execution, as with RNN, is the major benefit of this architecture.

However, according to literature, the overall training speed (convergence) of Transformer is relatively slower than RNN-based ASR. Thus, in this work, a joint transformer and CTC model, which facilitates faster convergence of the transformer model, is proposed. The reason for faster convergence lies in the fact that CTC explicitly aligned speech features and transcriptions, which allowed the sequence-to-sequence model to learn monotonous attention for ASR.

Most speech recognition systems use a limited set of words in their vocabulary. All words that are not part of the system's vocabulary are termed as Out Of Vocabulary (OOV) words. In morphologically rich languages like Amharic, there may be several OOV words. These words make the Amharic speech recognition task more challenging. The OOV problem arises because it is difficult to anticipate all the possible words that might be used by the speaker. Because of this, there are likely to be instances where the system encounters unexpected words. if the system encounters too many OOV words, its accuracy may suffer significantly. Therefore, building a diverse database of training data that includes a wide range of vocabulary and articulations will help to reduce the OOV problem in speech recognition tasks. However, due to lack of a significantly large Amharic speech corpus, it is difficult to build a diverse database of training data. Therefore, this research identifies suitable recognition units, such as characters and subwords, that are crucial to alleviating OOV problems.

1.2 Amharic Language

Amharic, the national language of Ethiopia, is a member of the Semitic language family and shares linguistic roots with other Semitic languages such as Hebrew and Arabic [2]. After Arabic, it boasts the greatest number of speakers in this linguistic family. Amharic's complexity lies in its diverse range of phonetic attributes, intricate morphological and syntactic structures, as well as numerous syllabic formations, all of which are reinforced by its unique writing system. To sum up, let us delve into each of these areas to gain a better understanding of the fascinating intricacies of Amharic.

1.2.1 Amharic Phonetics

Phonetics is the scientific study of speech sounds used in the language. This includes the description of the pronunciation of words, syllables and phonemes. Speech sounds are classified into vowels and consonants based on their acoustic characteristics [14]. The placements of the numerous articulators in the vocal tract and the condition of the vocal cords affect the characteristics of a sound. The classification of consonant sounds is based on three factors: voicing, manner, and location of articulation. We can categorize sounds into voiced and unvoiced based on the first factor. Stops, Fricatives, Nasals, Liquids, and Semivowels are the different types of sounds based on how they are spoken (manner of articulation). The locations of articulation include Labial, Dental, Palatal, Velar, and Glottal. Vowel sounds are, however, better specified in terms of the following three articulators: Position and height of the tongue, and rounding of the lips. In accordance to the above understanding of sound classification, the Amharic language has a total of 38 phonemes which are further divided into 31 consonants and 7 vowels [15]. Table 1.1 exhibits the phonetic representation of the consonants of Amharic as to their voicing, location and manner of articulation.

Table 1.1: Categories of Amharic Consonants

Manner of Articulation	Voicing	Place of Articulation				
		Labials	Dentals	Palatals	Velars	Glottals
Stops	Voiceless	ፕ[p]	ት[t]	ች[c]	ክ[k]	አ[ʔ]
	Voiced	በ[b]	ደ[d]	ጀ[j]	ገ[g]	
	Glottalized	ጵ[ṕ]	ጥ[t']	ጭ[č]	ቅ[q]	
	Rounded				ቈ[qʷ], ኰ[kʷ], ጐ[gʷ]	
Fricatives	Voiceless	ፍ[f]	ሰ[s]	ሸ[š]		ሀ[h]
	Voiced		ዘ[z]	ዠ[ʒ]		
	Glottalized		ጽ[ʃ]			
	Rounded					ኍ[hʷ]
Nasals	Voiced	ም[m]	ን[n]	ኝ[ñ]		
Liquids	Voiced		ል[l] , ር[r]			
Semi-vowels	Voiced	ው[w]			ይ[y]	

Source: Leslau et al. [14]

The consonants b, d, f, g, h, k, l, m, n, p, r, s, t, w, y and z in Amharic are phonetically transcribed the same as their English counterparts. However, there are specific sounds in Amharic that have special phonetic symbols but are similar to those used in English. These include 'ች' as 'ch' in 'church', 'ሽ' as 'sh' in 'shoe', 'ጅ' as 'j' in 'joke', 'ዥ' as 's' in 'pleasure' and 'ኝ' as 'ni' in 'onion'.

A distinctive feature of Amharic is the glottal stop 'እ', which corresponds to the negation sound 'uh-uh' found in English; however, it can be pronounced differently. According to sources, [14] states that the glottal sound may or may not be pronounced in Amharic, particularly when occurring between two vowels.

The Amharic language also contains sounds to which English does not have equivalents. These sounds, called glottalized sounds, include 'ጥ', 'ቅ', 'ጵ', 'ጽ' and 'ቍ', and each of them has a non-glottalized counterpart from the consonants 'ት', 'ክ', 'ፕ', 'ስ', and 'ች'. This variation in sounds adds depth and complexity to the Amharic language, making it an intriguing subject for linguistic study.

Amharic is also characterized by the existence of palatal consonants, which add to its unique phonetic features. In total there are six palatal consonants in Amharic - 'ች', 'ጅ', 'ቍ', 'ሽ', 'ዥ', and 'ኝ', each contrasting with corresponding dental consonants, 'ት', 'ድ', 'ጥ', 'ስ', 'ዝ', and 'ን', respectively.

Another phonetic feature that distinguishes Amharic from other languages is the marked pronunciation of long consonants, or "geminated consonants". Notably, the length of a consonant can have significant implications on the meaning of a word. A perfect example is that the difference between "ዋና(wana)" referring to "swimming" and "ዋና(wanna)" meaning "main or principal". According to [15], all Amharic consonants are capable of being geminated except two; 'እ' and 'ህ'.

Table 1.2 illustrates the placement of seven vowels, ኧ[ə], ኡ[u], ኢ[i], ኣ[a], ኤ[e], ኦ[I], ኦ[o], in articulation. As [14] notes, there is no precise correlation between Amharic and English vowels. The vowel ኧ[ə], for instance, is enunciated in the same manner as 'e' in 'bigger', while ኡ[u] is similarly pronounced as 'o' in 'who'. ኢ[i] and ኣ[a], on the other hand, have an 'ee' sound as in 'feet' and an 'a' sound in 'father', respectively. Finally, the other vowels ኤ[e], ኦ[I] and ኦ[o] are pronounced as 'a' in 'state,' 'e' in 'roses', and 'o' in 'nor', respectively.

Table 1.2: Categories of Amharic Vowels

Positions	Front	Center	Back
Low		አ[a]	
Mid	ኤ[e]	ኧ[ə]	ኦ[o]
High	ኢ[i]	ኣ[I]	ኡ[u]

Source: Leslau et al. [14]

1.2.2 Amharic Morphology

Morphology is the study that deals with the structure and production of words. It investigates the basic components of words and the rules that control their construction. Morphemes, the essential components of words, are examined as the building blocks of language. Amharic is a morphologically rich language in which words emerge from an intriguing root-pattern phenomenon. In this context, a root is a group of consonants (or radicals) with a common lexical meaning, whereas patterns are made up of intervening vowels that aid in the formation of word stems. A certain suffix or prefix added to a stem gives another stem [15]. For example, by intercalating the vowel ኧ[ə] and attaching the prefix አ[a] and the suffix -ኧ[ə] to the Amharic root fkr(ፍቅር) 'love', afəqərə(አፈቀረ) 'he loved' would be produced. Aside from this morphological trait, Amharic is renowned for its use of numerous affixes that generate inflectional and derivational word forms.

In Amharic, nouns, adjectives, stems, roots, and the infinitive form of a verb can be used to create new nouns by affixation and intercalation [16]. For example, from the noun səw(ሰው) 'person' another noun səwInət(ሰውነት) 'body'; from the adjective tIlIk(ትልቅ) 'big' the noun tIlIkInət(ትልቅነት) 'greatness'; from the stem sInIf(ስንፍ), the noun sInIfna(ስንፍና) 'laziness'; from root qld(ቅልድ), the noun qəld(ቀልድ) 'joke'; from infinitive verb məmItat(መምታት) 'to hit' the noun məmIca(መምቻ) 'an instrument used for hitting' can be derived. It is essential to keep in mind that as nouns are created, gender, number, definiteness, and case marker affixes inflect them.

Adjectives in Amharic can also be generated from nouns and verbal roots by adding prefixes or suffixes. For example, "dIngayama(ድንጋያማ)", an adjective that means "stony", derived from the noun "dIngay(ድንጋይ)", which means stone, while "zIngu(ዝንጉ)", meaning forgetful, is made by adding a suffix to the stem "zIng(ዝንግ)". Adjectives, like nouns, undergo inflection for gender, number, and case [15].

Verb derivation in Amharic is less common than noun and adjective derivation, although it does exist. Intercalation and affixation are required for the generation of new verbs. For example, from the root ”gdl(ግድል)”, meaning ”kill”, the perfective verb stem “gəddələ(ገደለ)” is obtained by intercalating the pattern “ə”. Passive and causative verb stems can also be derived from this, such as “təgəddələ(ተገደለ)” meaning “to be killed” and “asgəddələ(አስገደለ)” meaning “to cause death,” respectively, through the prefixes “tə-” and “as-”. Verbs in Amharic undergo inflection for person, gender, number, aspect, tense, mood, and negative markers, among other elements [15].

The foregoing explanation of Amharic morphology clearly demonstrates that Amharic is a language with an extraordinarily rich morphology. This specific language feature has a considerable influence on the development of Automatic Speech Recognition System (ASRS). This is due to the fact that it increases the size of the pronunciation dictionary and the rate of OOV terms, which raises the confusion of the LM. As a result of the increased prevalence of OOV words, dealing with Amharic ASR systems is more difficult. As a result, selecting an appropriate modeling (sub-word) unit is critical for efficiently managing these OOV difficulties.

1.2.3 Amharic Syllabification

Syllabification is the process of segmenting words, whether spoken or written, into syllables. A syllable consists of a vowel-like sound along with the surrounding consonants that are tightly linked to it. The essential constituents of syllables are Onset (the first phone in the series) and Rhyme (the remaining phone sequence), which contains nucleus (the centre peak of sonority) and Coda (the remaining consonants other than the onset) [5].

Amharic language is a syllabic language with Consonant-Vowel (CV) fusion in each grapheme, and all Amharic syllables don’t necessarily follow the CV sequence. Various researchers have investigated the syllable structure of Amharic language and developed different syllable patterns. According to [15], Amharic’s syllable structure is (C)V(C)(C), where C stands for a consonant and V for a vowel. Therefore, the types of syllables in Amharic include V, CV, CVC, VCC, VC, and CVCC.

A new syllabification technique has been proposed for Amharic in [17] by examining all six syllable types. This algorithm considers gemination and the unforeseeable nature of the Amharic epenthesis vowel. For instance, the word "ብስራት" (bsrat) meaning "Good news" does not show any vowel when transliterated to the International Phonetic Alphabet (IPA). However, through acoustic evidence, we can determine that there is an epenthetic vowel /I/; hence, its actual transliteration would be /bIsIrat/. Ultimately, gemination is a particular characteristic of the Amharic language that affects the semantics and syllabification of a given word. For example, the word "ይበላል" (yIbelall) meaning "he eats" might alter semantically when it's geminated as "ይበላል" (yIbbellall), which means "it will be eaten".

1.2.4 Amharic Writing System

Amharic is written from left to right and has its own set of characters known as fidal (ፊደል) that are used to form words in the language. The Amharic orthographic system is composed of 33 core symbols with each having seven different shapes or orders based on the associated vowels. Additionally, there are twenty labio-velars and eighteen labialized symbols, while the symbol ቨ[v], combined with its expanded seven orders, also forms part of the orthographic system.

Amharic writing falls into the category of syllabary because each character in the orthographic system represents a consonant and a vowel. Apart from the glottal stop and sixth-order symbols, the vowel has no independent existence [14]. The glottal stop consonant may or may not be pronounced, hence the symbol might represent a vowel in circumstances when it is not pronounced [18]. The sixth order symbols, on the other hand, just indicate a consonant, with the vowel እ[I] combined with consonants producing these symbols acting as an epenthetic vowel.

Although redundant orthographic symbols exist in Amharic that represents the same syllabic sounds, speech recognition efforts should focus on individual sounds rather than the orthographic symbols. For instance, four graphemes (ሀ, ሐ, ኀ, and ኸ) represent the "h" sound, while two graphemes (ሰ and ሠ) represent the "s" sound, two others (አ and ዐ) bring out the "a" sound, and the final two, (ጸ and ፀ), depict the "ts" sound with no known English equivalent.

After removing the redundant graphemes mentioned above, the Amharic orthographic system is reduced to 234 graphemes. Furthermore, both the first and fourth order graphemes of ሀ and ሐ represent the same sound, making only the first one necessary. The first order of the ኸ(h) sound is very distinct and must be taken into account, leaving a total of 233 distinct characters.

1.3 Problem Statement

Automatic Speech Recognition (ASR) is one of the most significant technologies and has a wide range of applications in modern life. For instance, disability assistance is an application that can be offered via the ASR system. An individual with hearing loss can use speech recognition technology to convert spoken words into text and then comprehend what is being said. Additionally, speech recognition makes it possible for those who have difficulty using their hands to use computers by speaking commands rather than typing.

So far, several attempts have been made to develop a continuous speech recognition system for Amharic using HMM [19, 20, 21, 22] and DNN [1, 3, 4] approaches. Authors [19, 20, 21, 22] employed the HMM-GMM paradigm with intermediate components to come up with an ASR system for Amharic language. Although they produced acceptable results in the past, the complexity of the HMM-GMM approach has substantially reduced the effectiveness of using these systems. The complexity is a result of the separate training of the language, pronunciation, and acoustic models.

Consequently, few Amharic speech recognition studies have concentrated on end-to-end modeling techniques such as Convolutional Neural Network (CNN) and RNN [1, 3], which seek to instantly simulate the translation between speech and labels without the need of intermediary components. Despite the fact that these approaches simplified the complexity of the conventional methods, the system's performance is hindered by the lack of an adequate Amharic speech corpus. In end-to-end approaches, a vast amount of data is needed to train a better model.

Despite producing results that are acceptable for the ASR task, both RNN and CNN models have drawbacks. RNN needs sequential iterations because the input is dependent on previous time steps. There is no way to parallelize the training process with this sequential computation. Hence, it is problematic with a longer input data sequence and takes a lot longer to train the model. CNN, on the other hand, transforms the frames simultaneously but only makes use of the local context in its receptive fields [13].

Although the recognition of continuous Amharic speech has made great strides in recent years, we still need to make more progress. For all published works, the key difficulties in developing Amharic speech recognition are the scarcity of available corpora. To come up with an enhanced Amharic speech recognition system, more Amharic speech and text corpora were used in this work. Furthermore, a joint CTC and attention-based model with the transformer architecture was utilized. This architecture removes recurrence and relies on its internal attention (self-attention) mechanism to make use of global context and determine relationships between input and output, which allows for significantly more parallelization. In addition, this research incorporates external LM, which improves Amharic speech recognition accuracy.

Research Questions

RQ1 How does joining Transformer and CTC models affect the Amharic speech recognition accuracy?

RQ2 How do character, subword and syllable recognition units affect the accuracy of Amharic speech recognition system?

RQ3 How does incorporating language model affects the accuracy of Amharic speech recognition system?

1.4 Objectives

1.4.1 General Objective

The general objective of this research work is to boost the accuracy of the continuous Amharic speech recognition system by leveraging more training data along with models based on joint Transformer and Connectionist Temporal Classification (CTC) with external language model integration.

1.4.2 Specific Objectives

- To implement and test transformer model
- To investigate the impact of using joint transformer and CTC based model for Amharic speech recognition

- To investigate the impact of incorporating Language Model (LM)
- To investigate the effects of different Amharic language recognition units such as characters, sub-words and syllables on Amharic speech recognition accuracy

1.5 Contribution

This research presents two significant contributions that aim to improve the accuracy of Automatic Speech Recognition (ASR) for Amharic. First, it proposes the integration of two state-of-the-art ASR techniques, namely CTC and Transformer joint training, which enables modeling of different Amharic language units (such as characters, subwords, and syllables) to achieve better ASR accuracy. This approach provides an effective solution to the long-standing issue of ASR in low-resource languages like Amharic.

Second, this research investigates the effectiveness of incorporating different language models into Amharic ASR, including character- and subword-based Recurrent Neural Network Language Model (RNNLM). These models help in the investigation of the effects of context-dependent and independent RNNLMs on end-to-end speech recognition models.

1.6 Scope and Limitation

A speech recognition system can be built to identify only read speech or to allow the user to speak spontaneously [23]. False beginnings, restarts, unfinished phrases, laughing, lip-smacking, coughing, and other such features are all aspects of spontaneous speech. The inclusion of spontaneous speech data during the training phase of a continuous speech recognition system helps to see spontaneous speech effects and enhances recognition accuracy. Although there is evidence that training speech recognition systems using spontaneous speech data provides more accurate results, an Amharic spontaneous speech corpus is currently unavailable. As a result, we focused on the Amharic continuous speech recognition system, which merely transcribes read speech into a set of letters or graphemes.

The research provides useful insights on identifying read speech, but it also has a few major limitations. One such limitation is that our research only focuses on Amharic recognition units like characters, subwords, and syllables. This limitation restricts the scope of the study to only a few potential recognition units, neglecting the possibility of comparing and contrasting other units like phonemes. Another limitation worth noticing is that the study does not allow for the use of punctuation marks, which are necessary for determining the kind of utterances. Finally, the exclusion of spontaneous speech limits the generalizability of the study's findings to real-life scenarios involving Amharic. Despite these limitations, the study's findings provide significant progress in advancing Amharic language speech recognition technology.

1.7 Document Organization

This document is broken down into five chapters. The first chapter covers the Amharic language in depth, with an emphasis on linguistic aspects relevant to automated speech recognition systems. It also comprises a statement of the problem and its rationale, the research's objectives, and the scope of the investigation. Existing Amharic speech recognition systems are reviewed in Chapter two. The methods and techniques we employed are detailed in Chapter three. In chapter four, we describe our experiments and results, as well as a comparison of Amharic speech recognition systems based on different language units such as character, syllable, and sub-word. Based on our findings, we arrived at the conclusions offered in Chapter five, which also incorporates some recommendations for researchers in this field.

Chapter 2

Literature Review

Automatic Speech Recognition (ASR) for Amharic was first studied when Birhanu [24] developed an isolated CV syllable recognition system in 2001. Since that time, a lot of efforts have been launched in academic study. The studies began with the use of small data sets produced by the researchers for their own research. The study in this area was greatly improved by the development of a moderately large speech corpus [25]. In this section, a brief overview of Amharic speech recognition studies that were conducted after the development of a medium-sized speech corpus by [25] is presented. Furthermore, we review an end-to-end ASR conducted based on the Transformer model for other languages.

Solomon et al. [19] presented the development of an ASRS for Amharic using limited available resources and the HMM methods. Since the acoustic samples of most words won't be seen during training, they took advantage of the segmentation of each word in the vocabulary into sub-units that occur more frequently and can be trained more robustly than words. In their study, CV syllables and tri-phone sub-units were utilized for acoustic modeling. For the language model, they trained word-based bi-gram language models using the Hidden Toolkit (HTK) statistical language model development modules. Since Amharic has a large number of morphologically rich words, some of them could be missing in the training set for the LM (OOV words issue). OOV words are the most challenging problems in this research work. Models with various HMM topologies have been developed in the work. The most accurate model was determined to have five states for each HMM and no skips. The context-independent syllable-based model performed somewhat worse in terms of accuracy when compared to a triphone-based model. The syllable-based recognizers, however, were discovered to be more efficient in terms of recognition speed and storage requirements. In light of this, they came to the conclusion that using CV syllables is a viable option for the development of Amharic speech recognition systems. However, the CV syllable unit does not address the issues of consonant gemination, irregular sixth order vowel realization, and glottal stop consonant realization, which have a significant impact on the accuracy of the sub-word transcriptions.

Martha et al. [20] looked into ways to alleviate the OOV issue by employing morphemes as both a lexical (pronunciation) and language modeling unit. This allowed them to demonstrate the impact of OOV words on the performance of an Amharic speech recognition system. It has been discovered that employing morphemes as dictionary entries and language model units significantly lowers the OOV rate and hence improves word recognition accuracy, especially for limited vocabularies (5k). However, the morpheme-based recognizers did not significantly boost word recognition accuracy as the morph vocabulary increased, which may be related to higher acoustic confusability and a constrained LM scope.

Martha et al. [21] presented the findings from their research on the usage of various units for acoustic, pronunciation, and language modeling for the low-resourced language Amharic. For acoustic modeling, triphone, syllable, and hybrid (syllable-phone) units have all been studied. In order to represent lexical and language models, words and morphemes have been studied. The hybrid AMs did not considerably enhance the word-based lexical and LM (i.e., word-based speech recognition). However, they resulted in a considerable reduction in the Word Error Rate (WER) for morpheme-based speech recognition. The syllable-based AMs also outperformed the triphone-based models in both word-based and morpheme-based speech recognition. This leads them to draw the conclusion that the optimal method for modeling Amharic speech recognition is to use morphemes in lexical and language modeling along with syllables and hybrid units in acoustic modeling.

Adey et al. [22] explored the idea of developing an Amharic continuous speech recognition system using the many syllable forms the language offers as acoustic units. The use of phones (including tri-phones) and CV syllables for acoustic modeling was the focus of previous studies in ASR for Amharic, as was mentioned above. Because of this, the potential benefits of employing Amharic syllables for acoustic modeling have not been fully explored. Baye Yimam [15] stated that the syllable structure of Amharic is (C)V(C)(C), where C represents a consonant and V a vowel. That means Amharic has the following syllable types: V, CV, CVC, VCC, VC, and CVCC. Thus, the study considered all six syllable types identified by [15] as the aim is to investigate the use of longer-length acoustic units for Large Vocabulary Continous Speech Recognition (LVCSR). Given sufficient training data, the findings of their studies indicate that it is feasible to employ all Amharic syllable types as acoustic units in LVCSR. Therefore, a speech corpus that fully encompasses all of Amharic's syllables is required.

The Amharic read speech corpus developed by Solomon et al. [25] was employed in all of the aforementioned research studies that we have so far assessed. It is a medium-sized speech corpus comprising just 20 hours of training speech, which was read by 100 training speakers in a total of 10850 different sentences. This is a fairly small amount of data to improve a speech recognition system.

Nirayo et al. [5] developed a novel corpus for the under-resourced Amharic language that is appropriate for training and evaluating speech recognition systems. The corpus prepared contains 90 hours of speech data from audio books and radio show archives with word- and syllable-based transcription. This corpus is then merged with the existing 20 hours of data, which contains varieties of speakers based on gender, age, and dialect. Researchers have so far used the HMM approach with intermediate modules for acoustic modeling to develop ASR systems for the Amharic language. In most cases, the hidden Markov and Gaussian mixture models (HMM-GMM) are used for the intermediate components. To avoid the need for an GMM and to ease the complexity of the approach, [5] explored the usage of a hybrid model, DNN combined with the HMM (DNN-HMM) model for acoustic modeling to develop an Amharic speech recognition system, and they demonstrated that the hybrid model has shown appreciable improvements over the HMM models on the same corpus. Moreover, a comparison of syllable and morpheme units for acoustic and language models is provided. In the paper, all six syllable templates, as well as the epenthesis vowel (I) and gemination, have been considered using the Amharic syllabification algorithm [17]. The syllable-based DNN-HMM model achieves a better syllable error rate than the syllable-based HMM model on the subset of the dataset. Despite the reductions in syllable error rate, developing a cutting-edge ASR system for Amharic was still a challenging and expertise-intensive task.

Nirayo et al. [1] worked on end-to-end speech recognition approaches for Amharic, which seek to instantly simulate the translation between speech and labels without the need for intermediary components. They investigated three models—Connectionist Temporal Classification (CTC), Convolutional Neural Network (CNN)s, and Recurrent Neural Network (RNN)s—in order to develop an end-to-end ASR system for Amharic, a language with a rich morphology but limited resources. The models were evaluated on roughly 52 hours of Amharic speech corpus (a subset of the dataset published on [5]). In order to discover a better recognition unit that may be utilized as an audio indexing unit for the Amharic end-to-end speech recognition system, they compared the grapheme, phoneme, and syllable-based end-to-end ASR systems for the language. Compared with the grapheme and phoneme-based models, the syllable-based models have shown better CER. As a result, they came to the conclusion that syllables make an effective recognition unit for Amharic speech recognition. The model was evaluated, and they found that the lowest CER and SER were, respectively, 19.21% and 39.98%, which are obtained after training 180 epochs. The result was obtained in the syllable-based 1D CNN model without the integration of lexicons (pronunciation) and language models. The recognition performance could even be boosted by incorporating more training data and integrating lexicon and language models.

Nirayo et al. [3] presented a well-designed 1-dimensional convolutional deep neural network architecture for low-resourced Amharic speech recognition. They investigated the use of convolutional neural networks with CTC under resource-restricted conditions. They extended the training set used by [1] from 52 hours to 70 hours and 46 minutes using the data level augmentation method. On the evaluation set, the Character Error Rate (CER) and Syllable Error Rate (SER) achieved are 18.38% and 27.71% respectively, without LMs integrated after training the networks for a maximum of 90 epochs. The result could even be improved by training for longer epochs, integrating an external syllable or character-based LM and producing more training data.

Eshete et al. [4] proposed hybrid Connectionist Temporal Classification (CTC) with attention end-to-end architecture and a syllabification algorithm for Amharic ASRS using its phoneme-based subword units. Subwords are sequences of characters, phonemes, and phonemes with an epenthesis vowel inserted by a syllabification algorithm. These subwords are generated by a Byte Pair Encoding (BPE) segmentation algorithm. In the work, the experiment results showed that a phoneme-based subword model with a syllabification algorithm and SpecAugment data augmentation technique was effective in achieving higher accuracy or a minimum WER in the CTC-attention end-to-end method. These subword models, which can represent longer contexts, are better able to address the OOV problem in the Amharic speech corpus than character-based methods do.

Zhou et al. [26] used Transformer as the basic architecture of a sequence-to-sequence attention-based model on Mandarin Chinese ASR tasks. In the work, a slight reduction in CER is achieved over the joint CTC-attention model by employing the Transformer model. Furthermore, they compared a syllable-based model with a context-independent phoneme (CI-phoneme)-based model. Their experimental results demonstrated that the syllable-based model with the Transformer performs better than its CI-phoneme-based counterpart on HKUST datasets.

Karita et al. [27] investigated the slower convergence of the Transformer model, namely its slower reduction in validation loss over wall clock time compared to RNN-based ASR. In the work, Transformer takes less time per iteration, but it takes many more epochs to converge. Transformer and RNN-based ASR accomplishments were combined by [27] to develop a faster and more accurate ASR system. A CTC with Transformer is utilized for co-learning and decoding in order to develop the model. Significant advancements in many ASR tasks are implemented by the ASR system. For instance, it reduced WER for the Wall Street Journal from 11.1% to 4.5% and for TED-LIUM from 16.1% to 11.6%.

Due to the limited corpus' availability and Amharic's intricate morphology, speech recognition in this language has proven to be particularly challenging. Thus, in this study, approximately 110 hours of the Amharic speech corpus developed by [5] were utilized in order to improve the accuracy of Amharic speech recognition. Furthermore, to improve the accuracy of the continuous Amharic speech recognition system, a hybrid transformer and CTC model with integration of an external language model was employed in this work. This research also investigates the impact of character, subword, and syllable units on the accuracy of the Amharic speech recognition system. As far as we can tell from our reading, no work has been published that employs the transformer-based end-to-end architecture for Amharic ASR tasks.

Chapter 3

Methodology

The methods and approaches for developing an Amharic speech recognition system are covered in this chapter. By thoroughly describing the dataset, feature extraction method, models, decoding techniques, recognition units, and evaluation metrics used in this work, our complete approach to developing an Amharic speech recognition system is demonstrated. This chapter provides a solid basis for the work's succeeding phases of execution.

3.1 Dataset

The primary source of data for technology development in the field of automatic speech recognition is the speech corpus (dataset). As a result, the first stage in developing an Amharic speech recognition system is to prepare an Amharic speech corpus. When producing a speech corpus, there are two options. The first option is to gather text corpora and invite language natives to read the text aloud while recording. The other option is to find and preprocess a range of previously recorded and transcribed speech for the development of speech recognizers.

In [25], the first option is used. The following method was used to gather the audio corpus: Texts were initially collected from a news website's archive and then divided up into sentences. Approximately 10,000 sentences have been chosen for the training set from the text database. The goal of choosing sentences from a text database is to create a collection that is both phonetically diverse and balanced in terms of the relative frequency of the sub-word units that will be modeled (phones, triphones, and syllables). Solomon et al. [25] selected sentences that contribute to the inclusion of all Amharic syllables in order to achieve phonetic richness. By choosing the sentences that support the preservation of the language's syllable distribution, the phonetic balance of the corpus is attained. Once the text corpus is available, recording the speech is an important next step in developing the speech corpus. An on-site recording under supervision was done in [25]. The recording was made in an office setting in Ethiopia using a laptop and a headset with a close-talking, noise-canceling microphone. The corpus includes 20 hours of training speech that was gathered from 100 speakers who read 10,850 sentences (28,666 words) in total. The speech recognition models will underperform due to a lack of training data, as this corpus is obviously smaller than other speech corpora that include hundreds and thousands of hours of speech data.

In [5], the second option is used. However, very few audiobooks and transcribed speeches were found, which limited the size of the corpus prepared. They have also made use of radio program archives that are openly accessible. Using Audacity's open-source tools, the audiobook and radio show archives have been segmented. The segmentation process was semi-automatic. Since most speech recognition toolkits expect relatively shorter utterances, the average length of the segments was 14 s. To align the text and spoken sentence, command-line tools and manual effort have been made. The preprocessing step involves fine-tuning, such as deleting non-speech components, removing lengthy silences, and fixing the audio samples using audio processing tools. Each subset's sampling frequency is standardized to 16 kHz, with sample sizes of 16 bits and monochannel bit rates of 256 kbps. Approximately 90 hours of speech corpus were prepared in [5] using word- and syllable-based transcription from audiobook and radio show archives. Finally, the corpus is merged with the 20 hours of data already compiled [25] to produce 110 hours of speech corpus, which includes a variety of speakers based on gender, age, and dialect. After aligning the text with the speech, numbers were converted into equivalent Amharic text as it is spoken in the recordings. Punctuation marks, foreign words, special letters, and symbols have all been removed, and abbreviations have also had their meanings extended manually.

It is expensive and labor-intensive to collect and prepare a very large speech corpus suitable for the development of a speech recognizer. In this study, the Amharic speech corpus prepared by [25, 5] was used, which comprises approximately 110 hours of speech. The dataset was partitioned into training and test sets, which contain 30,154 sentences and 3100 sentences, respectively. The dataset has both character- and syllable-based transcriptions for each utterance. To maintain variation, these speech corpora were gathered from a range of sources, including the Federal Negarit Gazeta, the Bible, fiction, sports, economic, and health news, as well as penal regulations.

3.2 Proposed Model Architecture

In this section, the proposed model architecture for Amharic speech recognition is presented. The architecture consists of five pivotal stages, namely feature extraction, subsampling, acoustic modeling, language modeling, and joint decoding, Figure 3.1.

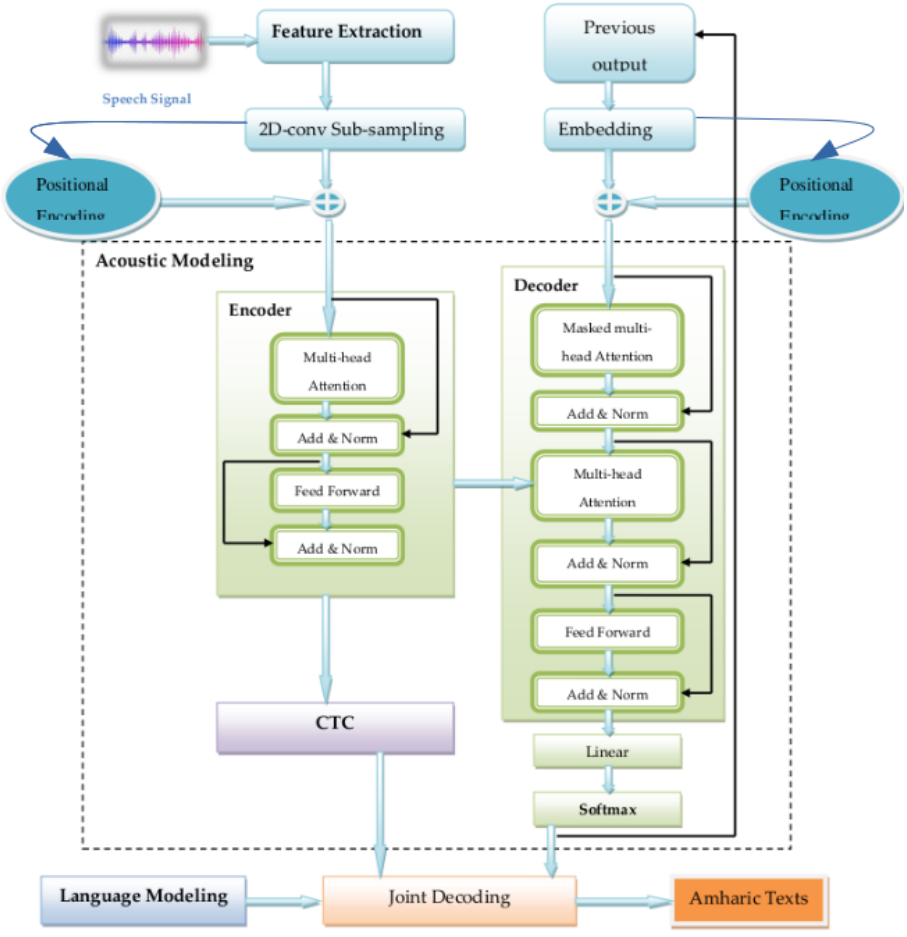

Figure 3.1: The proposed model architecture for Amharic speech recognition system

3.2.1 Feature Extraction

Feature extraction is the process of identifying the elements of a speech signal that can be used to distinguish linguistic content from background noise and information that is of no significance. The objective of feature extraction is to provide a compact representation of the input signal by computing a series of feature vectors. The standard method of feature extraction consists of the following steps:

3.2.1.1 Sampling and Quantization

The first phase in feature extraction is analog-to-digital conversion, which consists of two steps: sampling and quantization. Sampling is performed by obtaining samples of the speech signal at regular intervals at a certain rate called the sampling rate.

The next step is quantization, which entails lowering the number of bits used to represent each sample in the digital signal. One compelling reason to use this strategy is to reduce the amount of storage space and compute power necessary to process the signal. Linear Pulse Code Modulation (LPCM) is a popular quantization technique for encoding waveform samples in speech signal processing. This approach produces a sequence of quantized numbers that correspond to the amplitude of the signal at each sample point.

3.2.1.2 Windowing

Windowing is a technique for analyzing digitized, quantized speech signals. The approach divides these signals into smaller, equally-sized frames, after which a window function is applied to each frame. By dividing these continuous signals into smaller chunks, it is easy to analyze, capture crucial features more accurately, and reduce noise effects. Furthermore, windowing can help to reduce distortions that might occur when splitting signals suddenly. By preserving the continuity of all segments, overlapping windows allow for the reconstruction of the original signal.

The windowing process demands three fundamental parameters: the frame size or width of the window, the frame stride (also known as offset or shift) between subsequent windows, and the shape of the window itself [28]. In speech-related applications, typical window shapes include rectangular, hamming, and many more, each with specific qualities that make them especially suited for speech analysis.

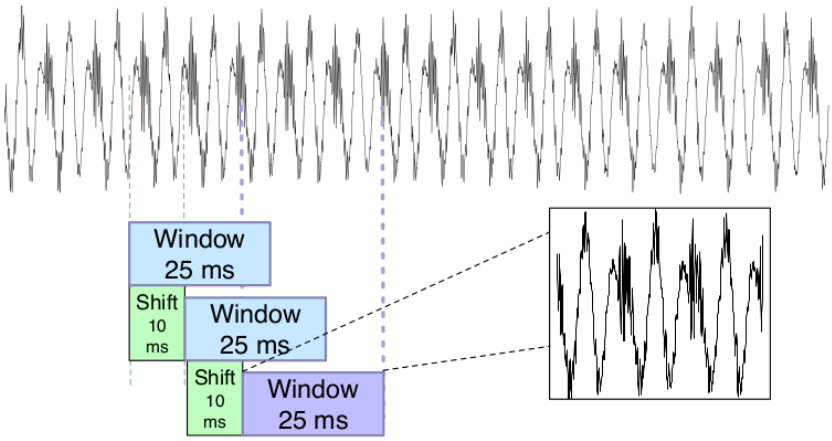

Figure 3.2: Windowing with a 10ms stride and a 25 ms rectangular window

Source: Jurafsky et al. [28]

The window shape in Figure 3.2 is rectangular; as can be seen, the extracted windowed signal appears exactly like the original signal. The rectangular window, however, suffers from the limitation of making discontinuous points, leading to high frequencies in the spectrum. Therefore, to keep the initial and final points in each frame consistent, each frame is multiplied by a hamming window. Both rectangular and Hamming windows are shown in Figure 3.3, and the equations are as follows (assuming a window L frames long):

$$RectangularWindow[n] = \begin{cases} 1, & \text{if } 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

$$HammingWindow[n] = \begin{cases} 0.54 - 0.46 \cos \frac{2\pi n}{L}, & \text{if } 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

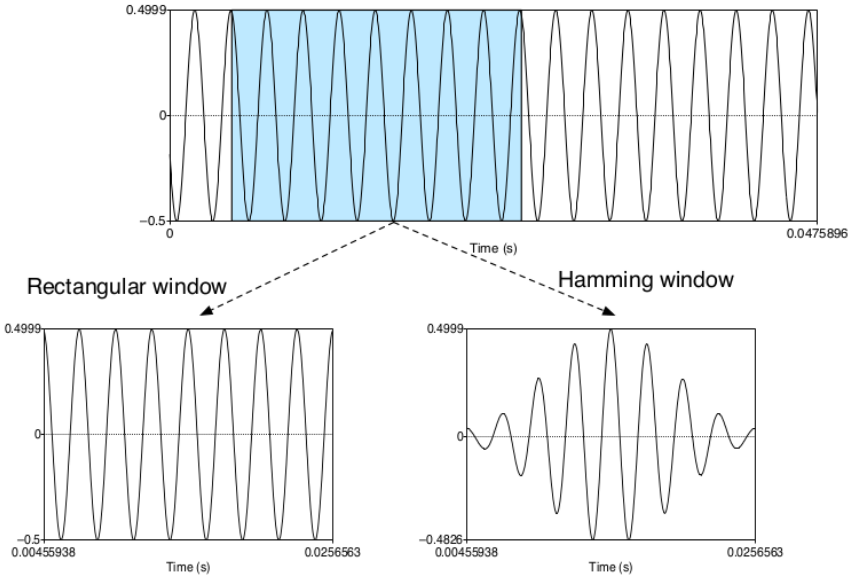

Figure 3.3: Windowing a sine wave with the rectangular and Hamming windows

Source: Jurafsky et al. [28]

3.2.1.3 Discrete Fourier Transform

In this step, spectral information is extracted from the windowed signal in order to estimate the energy distribution across different frequencies. A mathematical method called the Discrete Fourier Transform (DFT) is used to transform a time-domain signal sequence into a frequency representation. To obtain a complex spectrum, the DFT requires the input of a windowed signal. Squaring the magnitude of the resulting complex spectrum yields the power spectrum. The power spectrum depicts the energy distribution of the signal across different frequency bands. For instance, a 25ms chunk of a Hamming-windowed signal and its spectrum estimated by a DFT (with further smoothing) are displayed in Figure 3.4.

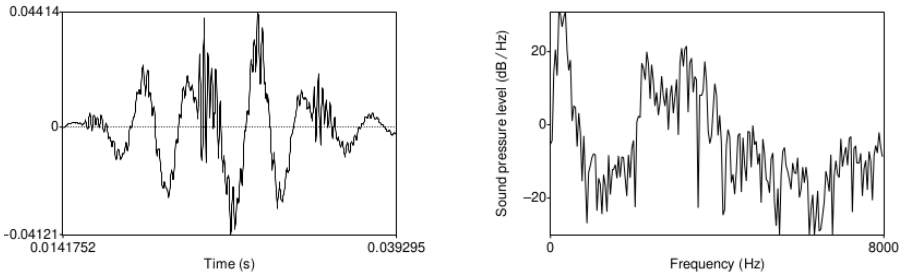

Figure 3.4: (a) A 25 ms Hamming-windowed portion of the vowel 'e' and (b) its spectrum computed by a DFT

Source: Jurafsky et al. [28]

3.2.1.4 Mel filterbank and log

The DFT provides information on the energy levels associated with each frequency band, but human speech perception is not a linear process [28]. The perceived distance between 100 Hz and 200 Hz is not the same as the difference between 1000 Hz and 1100 Hz. It is more sensitive to lower frequency ranges than to higher frequency ranges. Because low frequencies (such as formants) include essential information that distinguishes between distinct values or nasals, this phenomenon aids people in recognition, but high frequencies (such as stop bursts or fricative noise) are less significant for effective recognition. It is critical to model this aspect of human perception in order to improve the effectiveness of speech recognition systems. As a result, rather than collecting energies evenly at each frequency band, they should be collected according to the Mel scale, which seeks to produce a consistent perceptual scale of frequency that reflects the non-linear sensitivity of human hearing.

The DFT spectrum is then run through a bank of triangular filters evenly spaced along the Mel scale. These filters are designed to slightly overlap and capture information in multiple frequency bands. The energy values of the filtered outputs are added up for each overlapping section of the spectrum, and the logarithm of these values is calculated. A sample bank of triangular filters that exemplify this idea is shown in Figure 3.5, and the spectrum can be multiplied by the bank to create a mel spectrum. Finally, the logarithm of each mel spectrum value is calculated to account for the logarithmic human reaction to signal intensity. This method reduces the sensitivity of feature estimations to particular input variations, such as power changes caused by the speaker's lips moving closer or further away from the microphone.

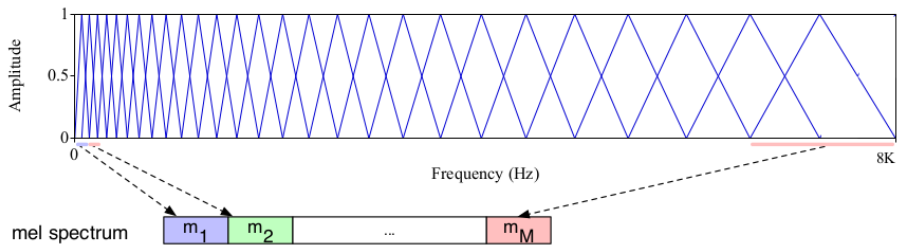

Figure 3.5: The mel triangular filter bank

Source: Jurafsky et al. [28]

3.2.2 Sub-sampling

The encoder-decoder architecture ideally suits scenarios where input and output sequences have vast differences in length [28]. For instance, a single word could consist of five letters and go on for around 2 seconds, equating to approximately 200 acoustic frames (at 10 ms per frame). Due to this significant length disparity between acoustic frames and letters, speech-based encoder-decoder architectures require a compression stage that shortens the length of the acoustic feature sequence before entering the encoder stage, Figure 3.6.

One popular method is using 2D-conv subsampling [29], which involves reducing the size of the input while preserving essential features. The 2D-Conv layer will train spatially invariant filters that can operate on a two-dimensional matrix of log-mel filterbank features as input. These filters slide over the input to provide a collection of convolved feature maps that capture different features of the input speech signal. Subsampling is then applied to reduce the output's dimensionality. Typically, this is accomplished by separating the input into non-overlapping sub-regions (e.g., 2x2 or 3x3) and employing an aggregation function such as max-pooling. Max pooling helps in the identification of key features in a region by reducing the influence of small differences such as noise or texture.

After subsampling, the acoustic feature frames F are transformed into sub-sampled sequence $X \in \mathbb{R}^{d^{sub} \times d^{model}}$ with 2D-CNN sampling layer. The d^{sub} is the length of the output sequence and d^{model} is the number of input feature dimensions to the Encoder. This process causes a high-level feature from the CNN extractor to enter the Transformer encoder's input.

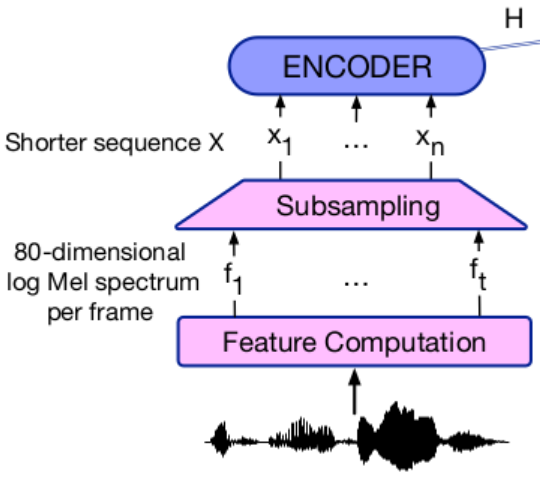

Figure 3.6: Schematic architecture that shows pre-encoder stages

Source: Jurafsky et al. [28]

3.2.3 Acoustic Modeling

Acoustic modeling of speech is the process of establishing statistical representations for feature vector sequences obtained from the speech waveform [30]. In most cases, neural networks are employed for this task. As part of efforts to enhance speech recognition systems for Amharic language, the Joint Transformer-CTC model is considered as an AM in this work. This model uses a Transformer encoder to generate a high-level features $h = (h_1, h_2, ..., h_L)$ for the input sequence $x = (x_1, x_2, ..., x_t)$ and subsequently applies both CTC model and Transformer decoder to simultaneously generate targets based on the high-level features.

The proposed system's specifics include detailed descriptions of the Transformer and CTC models in Sections 3.2.3.1 and 3.2.3.2, respectively. The joint Transformer-CTC objective will then be elaborated on in Section 3.2.3.3.

3.2.3.1 Transformer Architecture

Transformer is a contemporary sequence-to-sequence model that uses sinusoidal position information and a self-attention mechanism to completely do away with repetitions in typical RNNs [26, 31]. It is made up of one large block, which itself is made up of blocks of encoders and decoders. The primary function of the encoder model is to represent the input vector as a high-level representation. On the other hand, the decoder generates predictions one at a time. Consequently, at each time step, the model utilizes the high-level representation from the encoder model and previous predictions from the decoder as inputs for the current prediction.

Encoder

The encoder block comprises two important sub layers - a multi-head self-attention mechanism and a position-wise fully connected network. Each sub-layer produces an output, which is then passed through a layer normalization process. Additionally, the sub-layer input is directly connected to the output via a residual connection. The first encoder block receives the subsampled sequence input X.

Through the self-attention sub-layer, the X sequence is transformed into queries $Q = X \times W^q$, keys $K = X \times W^k$, and values $V = X \times W^v$. This transformation occurs using learnable weights, W^q and $W^k \in \mathbb{R}^{d^{model} \times d^k}$ and $W^v \in \mathbb{R}^{d^{model} \times d^v}$, where d^{model} represents the output dimension from the previous attention layer. Moreover, d^v, $d^k = d^q$ symbolize the dimensions of values, keys, and queries. A normalized weighted similarity Z is obtained from self-attention using softmax, which is showcased in the following equation:

$$SelfAttention(Q, K, V) = softmax(\frac{Q \times K^T}{\sqrt{d^k}}) \times V \tag{3.3}$$

To perform multiple attentions in parallel, Transformer uses multi-head attention (MHA). Multi Head Attention (MHA) comprises concatenating all self-attention heads at a specific layer.

$$MHA(Q, K, V) = [Z_1, Z_2, \cdots, Z_h]W^h \tag{3.4}$$

$$Z_i = SelfAttention(Q_i, K_i, V_i) \tag{3.5}$$

where h is the number of attention heads in a layer and i corresponds to the i^{th} head in the layer. The output obtained from MHA is normalized and fed into the Feed Forward (FF) sub-layer connected network. In the FF sub-layer, the input Z is applied to each position uniformly.

$$FF(z[t]) = max(0, z[t] \times W_1 + b_1)W_2 + b_2 \tag{3.6}$$

where z[t] represents the t^{th} position of the input Z.

Decoder

The decoder is very similar to the encoder, with MHA self-attention and completely connected sub-layers. However, the decoder incorporates a third component, the masked self-attention layer, which permits attention to be directed only to prior positions in the output sequence. The decoder provides a prediction $\hat{Y}$[t] at each time step based on the final encoder representation H_e and the previous target sequence Y[1:t-1]. To convert the decoder output into predicted probabilities for the next token, a common linear transformation and softmax function are used. The decoder employs MHA to establish such conditional dependency, allowing it to compute attention between encoder high-level features and previously decoded sequences. The decoder, like the encoder, has residual connections and layer normalization focused on each sub-layer.

Embeddings

Embeddings are a way to represent each token as a dense vector of d^{model} dimension. In Transformers, the initial vector representation starts as one hot encoding, but it is transformed into a dense vector through a trainable weight matrix before being passed through the network. This embedding method allows the model to learn semantic relationships between the tokens, allowing it to generalize better by understanding the context of each token in the sequence [13].

Positional Encoding

Positional Encoding (PE) is added to the token embeddings to indicate their position in the sequence, as self-attention does not have any notion of order or position [13]. It provides valuable information about the order of the words in the sequence for the model.

Transformers use sinusoidal PE with different frequencies as shown in equations below:

$$PE_{(n,2i)} = \sin \frac{n}{10000^{\frac{2i}{d^{model}}}} \tag{3.7}$$

$$PE_{(n,2i+1)} = \cos \frac{n}{10000^{\frac{2i}{d^{model}}}} \tag{3.8}$$

where n is the position of a word in the sentence and i is a position along the embedding vector dimension.

3.2.3.2 Connectionist Temporal Classification (CTC)

CTC is a technique that has revolutionized the way in which Transformers are trained. CTC leverages a unique approach that doesn't require any prior alignment between input and output sequences of varying lengths [32]. Instead, it presents a latent variable, known as the CTC path $\pi = (\pi_1, \pi_2, ..., \pi_L)$, to serve as the frame-level label of the input sequence.

The difficulty of speech recognition systems comes from the fact that there are many more input speech frames than there are output labels. For this typical need, a "blank" symbol is added as an extra label, and recurrence of labels is permitted to transfer the label sequence onto the provided CTC path, which has the same length as the input frames.

One of the most significant advantages of CTC over other methods is its ability to identify different paths that lead to a particular label sequence. By removing repetitions of the same label and blank symbols, CTC expands its mapping capabilities, providing richer and more accurate results.

After obtaining the encoder features from the transformer, CTC calculates the conditional probability of the label for each frame and assumes that the labels at different frames are conditionally independent [10]. So the probability of a CTC path can be computed as follows:

$$p(\frac{\pi}{x}) = \prod_{l=1}^{L} q_l^{\pi_l} \tag{3.9}$$

where $q_l^{\pi_l}$ denotes the softmax probability of outputting label π_l at frame l. $q_l = q_l^1, \cdots, q_l^{k+1}$ is often called the softmax output. The likelihood of the label sequence is the sum of probabilities of all compatible CTC paths:

$$p(\frac{y}{x}) = \sum_{\pi \in \Phi(y)} p(\frac{\pi}{x}) \tag{3.10}$$

where $\phi(y)$ denotes the set of all the CTC paths which can be mapped to the label sequence y.

A forward-backward algorithm can be employed to efficiently sum over all the possible paths. The likelihood of y can then be computed with the forward variable α_l^u and the backward variable β_l^u as follows:

$$p(\frac{y}{x}) = \sum_{u} \frac{\alpha_l^u \beta_l^u}{q_l^{\pi_l}} \tag{3.11}$$

where u is the label index and l is the frame index. The CTC loss is defined as the negative

log likelihood of the output label sequence:

$$L_{CTC} = -\ln(p(\frac{y}{x})) \tag{3.12}$$

By computing the derivate of the CTC loss with respect to the softmax output q_l, the parameters of the Transformer Encoder can be trained with standard back-propagation.

3.2.3.3 Joint Transformer and CTC

Aiming to leverage the strengths of both models, an approach can be taken to combine the CTC loss and transformer loss. Although CTC and transformer-based methods possess distinct benefits, they also exhibit their own limitations. While CTC assumes conditional independence between labels, thus requiring a potent external LM to account for long-term label dependencies, the Transformer attention mechanism uses a weighted sum over all inputs without constraints or guidance from alignments, resulting in difficulties when training the Transformer-based decoder.

It is noteworthy that the forward-backward algorithm employed in CTC can learn a monotonic alignment between acoustic features and label sequences [33]. This alignment can significantly accelerate the encoder's convergence. Furthermore, the transformer-based decoder can learn the interdependent relationship among target sequences. Thus, combining CTC and transformer loss not only promotes the transformer-based decoder's convergence but also permits the joint model to harness label dependencies effectively.

During Transformer training phase, the Transformer's decoder successfully predicts all label frames as $P_{Transformer}(Y/X)$, where Y denotes a ground truth sequence of the labels. The model effectively computes its training loss simultaneously in parallel, Equation 3.13.

$$L_{Transformer} = -\log P_{Transformer}(Y/X) \tag{3.13}$$

However, during the joint training stage, the approach taken differs from the traditional training method. Instead of restricting the model to Transformer alone, we adopt a multi-task loss method. This methodology aggregates both the CTC loss and the Transformer loss computed between the predicted label and actual label sequences. Therefore, the joint CTC-Transformer objective function is calculated as a weighted sum of CTC loss and Transformer loss, Equation 3.14.

$$L_{joint} = \lambda L_{CTC} + (1-\lambda) L_{Transformer} \tag{3.14}$$

where $\lambda \in (0, 1)$ is a tunable hyper-parameter.

3.2.4 Language Modeling

Language modeling is the technique of predicting the probability of a sequence of tokens in a particular language. It enables computers to understand natural language and respond in human-like ways.

The transformer model can implicitly learn an LM for the intended output domain via its training data. This data, which pairs speech with text transcriptions, may not be comprehensive enough to design an effective LM, emphasizing the need to locate vast amounts of appropriate text for proper model training. Therefore, in this study, the Amharic text corpus prepared by [34] was employed to develop a good LM.

This research investigates Long Short Term Memory (LSTM) based character level and subword level LMs for Amharic speech recognition tasks. LSTM neural networks are a form of RNN that can learn long-term dependencies in sequential input. They are especially valuable for language modeling because they may capture the context of a token as well as its influence on the tokens that follow in a sentence. The input to a typical language modeling task is a series of tokens, and the output is the probability distribution over the potential following tokens. The LSTM modifies its internal state depending on the current token and the prior state as it goes through the input sequence one token at a time. The LSTM's hidden state effectively retains the context of the sentence up to that point, allowing the model to predict the most likely next token.

In this study, word-level LM wasn't used since it is not suitable for Amharic. The morphological richness of Amharic is one of the reasons why word-level LM is unsuitable for it. Amharic, like many other languages, includes a large number of complicated word forms. Verbs, for example, can have various conjugations depending on tense, aspect, and person. This makes it difficult for a word-level LM to predict the following word in a sentence properly.

As an alternative, character- and subword-level LMs can be used in Amharic language modeling. Character-level models deconstruct each word into its constituent characters and predict the next character in the sequence. Subword-level models predict the next subword in a sequence by breaking each word down into segments called subwords. This method enables the LM to better represent the internal structure of Amharic words.

The evaluation of an LM's performance relative to another language model is measured by perplexity. This index estimates how many equally unique and probable words may follow each given word on average. The perplexity of the test set is a critical metric that dictates the difficulty of the recognition task. It is important to remember that perplexity tends to rise in tandem with the extent of the test set's vocabulary. Smaller vocabulary LMs, on the other hand, frequently experience problems due to a higher prevalence of OOV words. However, this issue may be mitigated by depending on character and subword-level models to significantly reduce the vocabulary size of the LM as well as the corresponding rates of OOV, thereby enhancing recognition accuracy.

3.2.5 Joint Decoding

In the decoding process, beam search is utilized to obtain the final selection of hypothesized sentences in the form of an n-best list. Each hypothesis score is calculated by joining the scores obtained from Acoustic Model (AM) and Language Model (LM) as shown in the equation below.

$$Yhat = argmax(\lambda \log \rho_{s2s}(Y/X) + (1-\lambda) \log \rho_{ctc}(Y/X) + \gamma \log \rho_{lm}(Y)) \quad (3.15)$$

Where, $\rho_{s2s}(Y/X)$ is the transformer decoder probability of the output sequence given the encoding feature sequence, $\rho_{ctc}(Y/X)$ is the CTC probability of the output sequence given the encoding feature sequence, $\rho_{lm}(Y)$ is the LM probability of the output sequence, λ and γ are hyperparameters named "CTC weight" and "LM weight" respectively.

3.3 Decoding techniques

Greedy search and beam search are well known decoding methods in a variety of sequence-to-sequence transformation tasks, such as ASR. These methods aim to generate the sequence outputs of tokens from a neural network model.

The model accepts the input sequence of tokens, where N is the number of tokens in the input.

$$X(inputsequence) = x_1, x_2, ..., x_N \quad (3.16)$$

The decoding methods generate the output sequence at each time step t, where T is the maximum number of tokens in the sequence.

$$Y(outputsequence) = y_1, y_2, ..., y_T \quad (3.17)$$

The probability of each output y_(t) is conditional of the previous token outputs. Mathematically, we represent it as it follows.

$$P(y_t|y_1, y_2, ..., y_{t-1}, x)$$

At each time step y_(t), the probability of each token in the vocabulary is computed. In consequence, the more tokens there are in the vocabulary, the more the computational cost increases.

3.3.1 Greedy search

Greedy search consists of taking the token with the highest conditional probability from the vocabulary V.

$$y_t = argmax_{y \in V} P(y|y_1, y_2, ..., y_{t-1}, x) \tag{3.18}$$

For example, we consider 14 words in our vocabulary V: <EOS>, <UNK>, አንድ, ቀን, ነው, ድርጊት, ሊከናወን, ነገር, ሂደትና, መልክ, በቻለ, ራሱን, የሚችል, ግን. Moreover, the maximum number of tokens in the sequence, T = 6.

	Time steps					
	1	2	3	4	5	6
<Eos>						
<unk>						
አንድ						
ቀን						
ድርጊት						
ሊከናወን						
ነገር						
ሂደትና						
መልክ						
በቻለ						
ራሱን						
የሚችል						
ግን						
ነው						

Figure 3.7: Greedy search algorithm

According to the Figure 3.7, the squares highlighted in red correspond to the words that have the highest conditional probability at each time step t. In the first time step the word with the highest conditional probability is "አንድ", in the second time step is "ነገር", etc. Therefore, the sequence output predicted by the decoder is: አንድ ነገር ግን በቻለ ሂደትና መልክ.

Main Drawbacks

Greedy search algorithm hides high probabilities that can be found in posterior tokens. Therefore, it does not always generate optimal output sequences.

3.3.2 Beam search

Beam search algorithm is the improved version of greedy search. Beam search has a parameter called beam width. The beam width is the number of tokens with the highest conditional probabilities at each time step t. In the Figure 3.8, the beam width=3.

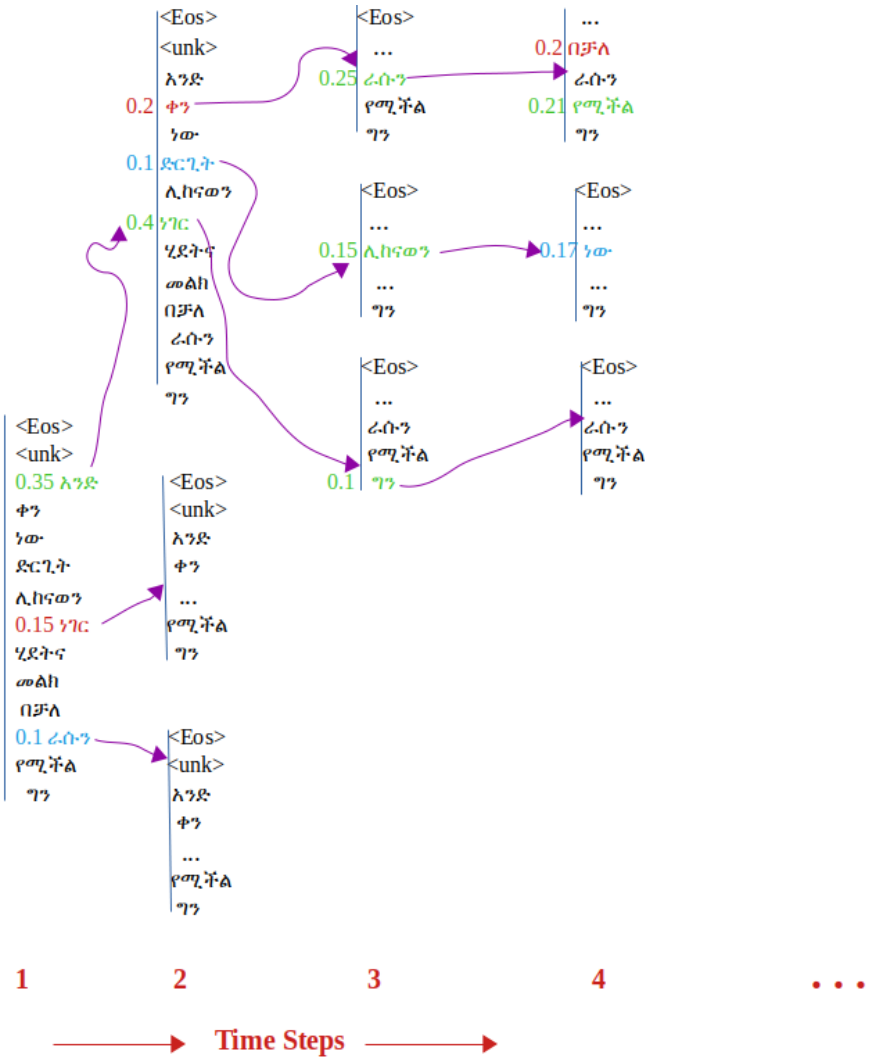

Figure 3.8: Beam search algorithm

Sequence 1 — አንድ ነገር ግን በቻለ — (0.35 * 0.4 * 0.1 * 0.2) = 0.0028

Sequence 2 — አንድ ቀን ረሱን የሚችል — (0.35 * 0.2 * 0.25 * 0.21) =0.0037

Sequence 3 — አንድ ድርጊት ሊከናወን ነው — (0.35 * 0.1 * 0.15 * 0.17) = 0.00089

Following the greedy search algorithm we selected the sequence 1, because the highest probability the greedy search found was in the second token (ነገር = 0.4), and it continues the token generation from this only branch. However, if we use beam search algorithm, the sequence with the highest probability is the sequence 2. This is a clear example of greedy search algorithm discarding sequences of tokens with higher probability.

Drawbacks

- Increasing the beam width, the quality of the output sequence improves significantly, but it reduces the decoder speed.
- There is a saturation point, where even if we increase the number of beam width, the quality of the decoding does not improve anymore [35].
- It does not guarantee of finding the output sequence with the highest score [36].

3.4 Fundamental Amharic speech recognition units

An end-to-end speech recognition model has the ability to directly map acoustic frames to label sequences, which are made up of characters, subwords, and words. Since Amharic is a morphologically rich language, a word-based end-to-end model suffers from the OOV words problem. Although a large vocabulary can be used to solve this problem, the model suffers from high computing costs as a result of its large softmax output. The OOV words can't be recognized since they will be unavailable during the training of the model. Moreover, it would be impracticable to train models for each word in the Amharic ASRS lexicon, which comprises thousands of words, many of which are rare. Therefore, it is crucial to segment speech into smaller segments rather than whole words in order to build Amharic speech recognition systems. Furthermore, by breaking down words into smaller subunits, words that are not part of the training sample can still be recognized.

In this study, words from the vocabulary have been segmented into syllables, characters, and subwords. A syllable is a unit of sound composed of one or more vowels and consonants. Syllables have a crucial role in how stress and intonation are spoken. The syllable structure of Amharic is (C)V(C)(C), where C means consonant and V denotes vowel [15]. Types of syllables in Amharic include V, CV, CVC, VCC, VC, and CVCC. For instance the text: ”ነገር ግን አንድ ቀን ራሱን በቻለ ሂደትና መልክ ሊከናወን የሚችል የማይቀር ድርጊት ነው” can be decomposed into a sequence of syllable unit: ”nə gər gIn and qən ra sun bə ca lə hi də tI na məlk li kə na wən yə mi cIl yə ma yI qər dI rI git nəw”.

A character in Amharic is a fundamental building block of written language that visually represents a sound or meaning. Character is a suitable unit for end-to-end speech recognition since every text can be easily separated into character sequences. However, utilizing characters would make understanding long-term (word-level) context dependencies more difficult. The difficulties posed by character units might be overcome by the use of subword units. A subword is a group of characters that together create a coherent phrase that is less than a word. These words, which usually include stems, roots, prefixes, and suffixes, serve as crucial linguistic building blocks to produce words. BPE algorithm forms subwords using a string of characters, where the most frequent character combinations are merged [32]. The algorithm consists of the following steps:

1. Initialization - Initialize a vocabulary V with all unique characters in the training data and an additional symbol ”_”, indicating the end boundary of a word.

2. Create character pairs - Count all adjacent character pairs in the vocabulary and store them in a dictionary, D.

3. Merge most frequent pairs - Sort the dictionary D by frequency of occurrence and select the most frequent pair (X, Y). Replace all occurrences of (X, Y) with a new symbol XY in the vocabulary, V.

4. Update vocabulary - Add the new symbol XY to the vocabulary, V, and repeat steps 2-4 until we reach a predetermined number of subwords or until we no longer find frequent pairs.

For example, let's consider the text "ድንበር መስበር". We initialize the vocabulary with all unique characters in the training data plus an underscore symbol: ድ, ን, በ, ር, መ, ስ, _. Note that an underscore represents the end boundary of a word as one character. The first step creates all possible pairs of characters in the vocabulary: (ድ, ን), (ን, በ), (በ, ር), (ር, _), (_, መ), (መ, ስ), (ስ, በ), (በ, ር). Next, we count the frequency of each pair in the training data and merge the most frequent pair, (በ, ር), replacing it with the subword "በር". We then update the vocabulary: ድ, ን, መ, ስ, _, በር. We repeat the process until we have reached a predetermined number of subwords or until there are no more frequent character pairs.

The resulting vocabulary contains subwords that represent common pairs of characters in the training data, allowing for more effective tokenization and better modeling of rare or unseen words. In this study, 500 as well as 20000 sub-word units were prepared just to see their impacts on recognition accuracy.

3.5 ASR evaluation metrics

There are different types of evaluation metrics in ASR, including Word Error Rate (WER), Character Error Rate (CER), Phoneme Error Rate (PER), and Syllable Error Rate (SER). These metrics measure how many tokens are incorrect in the full transcript of a speech recognition system. They are calculated by dividing the total number of errors in the hypothesis sequence (insertions, deletions, and substitutions) by the total number of words, characters, phonemes, or syllables in the reference or target sequence. Equation 3.19 - 3.22 shows the formula for computing CER, WER, PER, & SER, respectively.

$$CER = \frac{(S + D + I)}{N_c} \times 100\% \tag{3.19}$$

$$WER = \frac{(S + D + I)}{N_w} \times 100\% \tag{3.20}$$

$$PER = \frac{(S + D + I)}{N_p} \times 100\% \tag{3.21}$$

$$SER = \frac{(S + D + I)}{N_s} \times 100\% \tag{3.22}$$

Where S represents the number of substitutions, D represents the number of deletions, I represents the number of insertions, N_c, N_w, N_p and N_s represents the total number of characters, words, phonemes and syllables, respectively, in the target sequence.

3.6 Summary

This chapter explains the dataset used and the proposed methodology for the Amharic speech recognition system, including the fundamental recognition units such as character, subword, and syllable. The proposed approach entails feature extraction to transform the speech waveform into acoustic feature vectors, followed by subsampling. Feature vectors along with positional encoding information are provided to the transformer encoder, which turns the feature vectors into a high-level representation. The transformer decoder takes the high-level representation and predicts the output label sequence based on previous predicted labels, while CTC also predicts the output label sequence. Additionally, a Language Model (LM) is incorporated into the Acoustic Model (AM) to improve Amharic ASR accuracy. During inference, the sum of log probabilities is taken from the LM and AM to find the best hypothesis. ASR evaluation metrics are also discussed in this chapter.

Chapter 4

Result and Discussion

This chapter explores the precise details of the experimental setup and the findings. The effect of applying the hybrid Transformer and CTC model is presented in this chapter. By carefully analyzing our studies, we were able to explain the effects of different Amharic language recognition units, such as characters, sub-words, and syllables, on the accuracy of Amharic speech recognition. We were also able to demonstrate the effect of incorporating a language model into our system. Through an in-depth analysis, we have gained valuable insights into the underlying factors that influence Amharic speech recognition accuracy, laying the foundation for future work in this field.

4.1 Experiment Setup

Google Colab, a convenient cloud-based service provided by Google, was used for our experiment's training and testing phases. The Espnet [37], an end-to-end speech processing toolkit with a Pytorch backend was employed, to achieve the results we obtained in this speech recognition system. The transformer model is a highly sophisticated aspect of this system, consisting of twelve encoder layers and six decoder layers that form a 2048-dimensional feed-forward network. In order to increase the attentiveness of our system, eight attention heads, each with 512 dimensions were also incorporated. The neural network was chosen because of its accuracy and speed, which were improved by initializing the transformer weights with the Xavier uniform method.

A multi-task loss weight of 0.3 was used to implement the joint training method. In addition, several regularization techniques such as 10% dropout on every attention matrix and weight in Feed Forward (FF), layer normalization before every MHA and FF, as well as label smoothing with a penalty of 0.1 were used to prevent overfitting on the training set.

The training was conducted for 100 epochs using Pytorch backend and a batch size of 8. Further, the Noam optimizer was utilized with warmup steps, label smoothing, gradient clipping, and accumulating gradients to train the proposed speech recognition system.

A sampling rate of 16000 Hz was used to create audio frames with intervals of 25 ms and 10 ms for the feature extraction process, which produced 80-dimensional log mel-filterbank features that were used for both training and decoding. To ensure robust results, the feature vectors were normalized using a global cepstral mean and variance normalization technique.

The right parameters must be chosen in the area of language modeling based on the language units being employed. To this goal, the sub-word units and character units of two different types of LMs were investigated. With sub-word LM, a 2-layer LSTM architecture with 1024 hidden units, Noam optimization, 64 batches, and a maximum sequence length of 55 was employed. Alternatively, character LM employed a 4-layered LSTM architecture, with each layer containing 512 hidden units. Like the sub-word model, the character LM also utilized Noam optimization to enhance performance. This model's batch size was increased to 256 because character-level modeling often has longer sequences. The maximum sequence length was set at 400 characters. In this work, both sub-word and character LMs were trained for 100 epochs, during which the neural network's weights and biases were adjusted iteratively to converge to an optimal result.

4.2 Experiment Results

4.2.1 Investigation of joint Transformer and CTC model

In this study, we first conducted training on our baseline implementation of RNN and joint RNN-CTC for ASR. Subsequently, training was then carried out on the Transformer and joint Transformer-CTC models. The results in Figure 4.1 show that there are notable differences in the training losses of RNN, joint RNN-CTC, Transformer, and joint Transformer-CTC models over the course of several epochs.

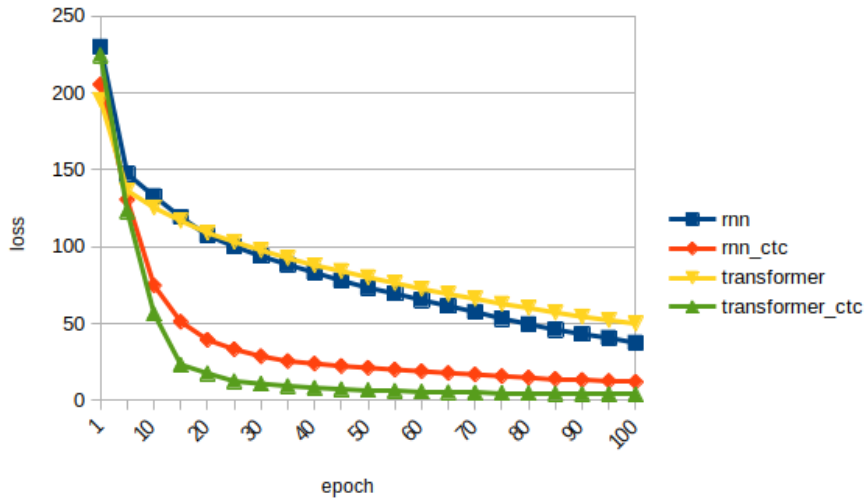

Figure 4.1: Training losses in character-based AM

First, it is observed that both RNN and Transformer models do not achieve losses close to zero throughout the training process. This suggests that these models experience slower convergence compared to the joint models. The slower convergence implies that the models require more epochs to attain lower losses and reach a certain level of accuracy. However, when comparing the RNN and Transformer models, it is noted that the RNN model exhibits a lower loss compared to the Transformer model after an extended number of epochs. This indicates that the RNN model performs better in terms of reducing the training loss as more training iterations are performed. It is important to note that this comparison is specific to the training loss criterion employed in this study.

On the other hand, the joint RNN-CTC and joint Transformer-CTC models demonstrate a faster convergence compared to their respective non-joint counterparts. This suggests that incorporating the CTC component in both the RNN and Transformer models aids in achieving quicker convergence. This is due to the fact that CTC explicitly aligns speech features and transcriptions. The alignment helps the Transformer model to learn monotonic attention for ASR. Monotonic attention refers to the output sequence produced by explicitly attending the input sequence from left to right. In addition, this attention mechanism restricts the decoder to only attend to parts of the input sequence that it has not attended to before. In other words, the decoder will only be allowed to attend to the parts of the input sequence that come after the regions already attended to. Therefore, attention is only allowed to move forward monotonically. This approach of monotonic attention is important for the model to be able to converge much more quickly and effectively as a result of quick attention computations as well as improved decoding efficiency.

Moreover, among all the models considered, the joint Transformer-CTC model exhibits the fastest convergence. This implies that the combination of the Transformer architecture with the CTC component is highly effective in accelerating the convergence of the model. The joint Transformer-CTC model achieves lower losses at a quicker speed compared to the other models evaluated in the study.

In summary, the results suggest that while the RNN and Transformer models show slower convergence, the RNN model performs slightly better than the Transformer model regarding reducing the training loss over a greater number of epochs. However, the joint RNN-CTC and joint Transformer-CTC models demonstrate faster convergence, with the joint Transformer-CTC model being the most efficient in terms of achieving lower losses quickly. These findings underscore the importance of considering different model architectures and incorporating additional components like CTC to enhance convergence and improve performance in sequence-to-sequence tasks.

4.2.2 Investigation of Language Model

In this work, 368,394 sentences extracted from the Contemporary Amharic Corpus (CACO) [34] have been used to derive the vocabulary for the dictionary and to train both character- and subword-level LMs.

The performance evaluation of an LM is measured by perplexity. The perplexity of the test set is a critical metric that dictates the difficulty of the recognition task.

The validation perplexity decreased alongside the training perplexity, as shown in Figures 4.2 and 4.3 below. Both the training and validation perplexities have decreased, which indicates that neither model overfits the training set and should perform more reliably on new data. The perplexity for character- and subword-level LMs, respectively, on unseen test data is 6.35 and 10.24. As can be seen, the subword-level LM has a higher degree of perplexity than the character-level LM. This is because perplexity tends to increase as the vocabulary size of the test set increases. It is important to note that the models should perform better at predicting new data if they achieve lower test perplexity.

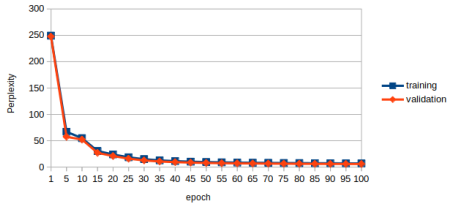

Figure 4.2: Training and Validation perplexities of character-level LM

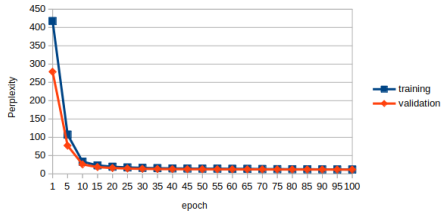

Figure 4.3: Training and Validation perplexities of subword-level LM

4.2.3 Impact of joint decoding

Our research findings have been categorized into three main categories: character-based AM with character- and subword-level LMs; subword-based AM with character- and subword-level LM; and syllable-based AM. These categories allow for a thorough analysis of the results because they accurately classify the experiment results as well as represent the different methods used.

4.2.3.1 Character-based acoustic model

The character-based AM was evaluated for its ability to decode test data, with the overall goal of reaching maximum recognition accuracy. The evaluation utilized two commonly used metrics: CER and WER.

The model's error rate is shown in Table 4.1. The best preliminary results show a CER of 9.53% and a WER of 28.03%. To improve the model's accuracy, different LMs such as character-level LM and a subword-level LM were incorporated into the decoding process.

Acoustic model	Language model	Character error rate (CER)		Word error rate (WER)	
		Greedy decoding	Beam search decoding	Greedy decoding	Beam search decoding
Character	-	10.43%	9.53%	29.84%	28.03%
Character	Character-level LM	-	8.84%	-	27.61%
Character	Subword-level LM	-	9.94%	-	25.8%

Table 4.1: Decoding results of character based acoustic model

The CER dropped to 8.84% after integrating the character-level LM, demonstrating an improvement in recognition of individual characters. The WER, on the other hand, improved, falling from 28.03% to 27.61%. This was an unexpected outcome because the character-level LM doesn't offer the capability of modeling longer context dependency to recognize complete words. The employment of the subword-level LM, on the other hand, resulted in a higher CER of 9.94% but a lower WER of 25.8%. This finding revealed that by taking into consideration the expected combinations of characters inside each word, the subword-level LM was able to increase the model's ability to recognize complete words.

It is worth mentioning that the discrepancy in error rates between the character-level and subword-level LMs can be traced to their fundamental properties. A subword-level LM, in particular, is more adapted to capture the most likely sequence of characters that constitute a word, which can enhance overall WER. A character-level LM, on the other hand, is more effective at improving CER because it focuses on predicting the probability of specific characters. The study also found that both character- and subword-level LMs are more successful at managing OOV words, which can be frequent in particular datasets and result in greater word error rates.

4.2.3.2 Subword-based acoustic model

The performance evaluation of a trained subword-based joint transformer and CTC model is shown in Table 4.2, along with the impact of utilizing a character- and a subword-level LM on the model's performance. Like the character-based AM, CER and WER metrics are used to evaluate the subword-based AM.

The study's best preliminary result indicated that a model with 500 subword units extracted from the training texts had a CER of 10.02% and a WER of 26.67%. This finding show that the subword-based AM achieved a lower WER and a higher CER as compared with the character-based AM. The research also investigates the impact of using different number of subword units. When the model was trained with 20000 subword units, the CER increased significantly to 28.42% and the WER also increased to 46.3%. This revealed that the high degree of complexity caused by additional subword units creates overfitting, causing the model to become unable to generalize successfully on new unseen data. Furthermore, increasing the number of subword units results in fewer unit occurrences in the text corpus. For instance, the least frequent unit for the 500 subword set was "ስፉ" which appeared 984 times in our training data, whereas for the 20000 subword set, the least frequent unit was "_ይረክ" which appeared only six times. This problem of data sparsity has the effect of degrading model performance.

Acoustic model	Language model	Character error rate (CER)		Word error rate (WER)	
		Greedy decoding	Beam search decoding	Greedy decoding	Beam search decoding
Subword-500 unit	-	10.77%	10.02%	28.53%	26.67%
Subword-20000 unit	-	30.21%	28.42%	49.6%	46.3%
Subword-500 unit	Character-level LM	-	9.79%	-	26.02%
Subword-500 unit	Subword-level LM	-	9.99%	-	24.61%

Table 4.2: Decoding results of subword based acoustic model

Character- and subword-level LMs have been incorporated to increase the model's performance. The findings showed a considerable increase in accuracy when the character-level LM was used, with CER decreased to 9.79% and WER lowered to 26.02% when 500 subword units were used. The character-level LM offered extra information on the sequence of characters that are expected to occur in the text, allowing the model to make more accurate predictions. Incorporating a subword-level LM, on the other hand, resulted in even better performance, with a WER of 24.61%. The subword-level LM caught the intricate patterns of subwords seen in the text and offered longer context to the model, allowing it to make more accurate predictions.

The most important finding of the study indicated that subword-based joint transformer and CTC models with fewer subword units outperform models with more subword units. Choosing an optimal number of subword units is critical to prevent having an extremely complicated model, which leads to overfitting, as previously described.

4.2.3.3 Syllable-based acoustic model

The goal of this experiment was to assess the performance of a Syllable-based Joint Transformer and CTC model for Amharic speech recognition. Table 4.3 depicts the performance evaluation of a trained syllable-based acoustic model. The key performance indicators employed for assessment were Phoneme Error Rate (PER) and Syllable Error Rate (SER), which evaluate the accuracy of detecting individual phonemes and overall syllable structure, respectively.

The findings show that the proposed approach is very successful for the problem of Amharic speech recognition. The model, in particular, obtained an acceptable PER of 7.05%, showing that it can reliably distinguish the majority of distinct phonemes. Furthermore, a SER of 13.3% was found, indicating that the model is quite good at detecting the overall syllable structure in continuous speech.

Acoustic model	Phoneme error rate (PER)		Syllable error rate (SER)	
	Greedy decoding	Beam search decoding	Greedy decoding	Beam search decoding
Syllable	7.56%	7.05%	14.88%	13.3%

Table 4.3: Decoding results of syllable based acoustic model

To grasp the relevance of these findings, it is important to keep in mind that syllables and phonemes are the two basic units of speech, with syllables being larger and comprised of one or more phonemes. While both PER and SER represent errors made by the model, they measure different aspects of its performance. SER considers not just individual phoneme errors but also syllable boundaries and overall syllable structure, giving it a more relevant metric for continuous speech recognition tasks involving longer speech sequences.

Although the results are acceptable, more research is needed to find possible areas for improvement. The model's significantly higher SER, for example, shows that more training data may be required to properly capture the different aspects influencing overall syllable structure, such as tone and stress patterns in speech. Modifying the model architecture and/or tuning hyperparameters, on the other hand, may also enhance performance and minimize errors.

In general, a PER of less than 10% and/or a SER of less than 20% is regarded reasonable performance for speech recognition models. However, while analyzing the findings, the specific requirements of the task should be considered. Based on the findings of this study, it is possible to infer that the proposed approach has significant promise for the task of Amharic speech recognition and needs further investigation.

4.2.4 Training and inference time

Figure 4.4 illustrates the training and inference time of the joint Transformer-CTC and RNN-CTC models. The training time for the joint RNN and CTC model can be relatively high due to RNN's sequential nature, processing input data step-by-step. On the other hand, joint Transformer and CTC model generally has faster training times as Transformer can process input data in parallel. Similarly, when it comes to inference, Transformer-CTC model usually outperform RNN-CTC model in terms of speed. This is because Transformers can also process input sequences in parallel, making them more efficient for inference on modern hardware like GPUs.

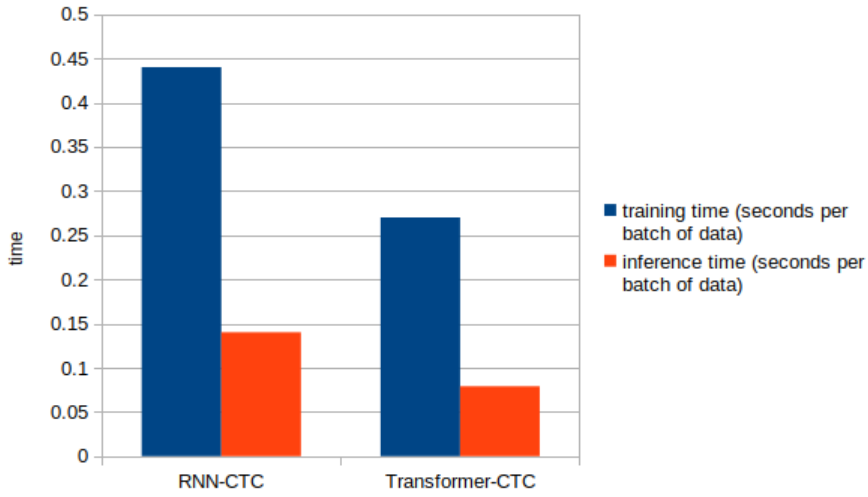

Figure 4.4: Training and inference time of Transformer-CTC and RNN-CTC

4.3 Discussion

In this work, the effectiveness of the joint Transformer and CTC model was evaluated using the greedy decoding and beam search decoding method. In contrast to beam search decoding, greedy decoding chooses the token with the highest probability at each timestep without taking into account the impact of subsequent tokens. Our results showed that beam search decoding outperformed greedy decoding in all experiments, showing that it is more effective at decoding speech recognition hypothesis sequences. Our findings specifically show that beam search is a more workable method for enhancing speech recognition accuracy than greedy decoding.

Characters, syllables, and character-based subwords were used as recognition units to assess the effectiveness of our proposed transformer-CTC end-to-end ASR for Amharic. In this work, character-based and subword-based end-to-end ASR models' CER and WER were compared. The evaluation's findings showed that when character error rates are our main concern in ASR, character-based acoustic models perform better than subword-based acoustic models. This can be attributed to the fact that character-based models are easier to understand because each character represents a different language sound, and as a result, they typically perform better when recognizing single characters. On the other hand, subword-based models might outperform character-based models when the primary objective of ASR is to minimize word error rates. This is due to the fact that subword-based models are more adaptable in encoding frequent sound combinations, which is especially helpful in languages like Amharic that have a variety of complex sounds. Subword models improve the system's ability to recognize complex sound combinations, making it simpler to identify words made up of these complex sound combinations.

The three research questions mentioned in Chapter one are answered as follows:

RQ1 How does joining Transformer and CTC models affect the Amharic speech recognition accuracy?

The transformer model used in our experiment failed to converge and did not generalize well to the unseen test data in our investigation. There could be several reasons behind the lack of convergence and poor generalization of the transformer model trained for the Amharic speech recognition system.

One possible reason is insufficient training data; it is likely that the provided training data is insufficient to capture the variability of the language, particularly if the model has a high number of parameters. This might result in overfitting, a condition in which the model only memorizes the training data but fails to generalize to new data. This problem can be solved by collecting more training data or augmenting the existing data with techniques such as noise injection, speed perturbation, or pitch shifting. The other possible reason is inappropriate model architecture: the transformer model may not be suitable for Amharic speech recognition, or it may need to be modified to better capture language characteristics such as tone fluctuations, prosody, or vowel harmony.

Our findings show that joining the Transformer and CTC models has a positive impact on reducing overfitting and improving the convergence speed. Furthermore, in our research on the low-resource Amharic speech recognition system, the joint Transformer-CTC model significantly outperforms the pure Transformer model.

RQ2 How do character, subword and syllable recognition units affect the accuracy of Amharic speech recognition systems?

The accuracy of Amharic speech recognition can be significantly impacted by the use of different recognition units. The different speech components used to build the system, such as characters, subwords, or syllables, are referred to as different recognition units.

Our study shows that for Amharic speech recognition with limited resources, subword recognition units tend to perform marginally better than character-based units. However, the most efficient method may vary depending on the precise data at hand, so it's important to test using more training data. Moreover, our results depict that the syllable-level recognition model works well for Amharic.

RQ3 How does incorporating language model affects the accuracy of Amharic speech recognition system?

Incorporating a language model significantly improve the accuracy of an Amharic speech recognition system. The language model provides contextual information on the most likely sequence of tokens in Amharic, which improves the recognition accuracy by biasing the speech recognition system to select the more probable tokens when there is ambiguity in the acoustic signal.

The language model units, which could be characters or subwords, help the speech recognition system to better understand and interpret the words being spoken by the user. As the system processes each audio input, the language model uses statistical analysis to suggest potential character or subword combinations that may match the incoming audio.

A character-level language model works by considering individual characters in a word and their likelihood of appearing together based on the training data, which can help the system better recognize out-of-vocabulary words.

A subword-level language model uses small units of words (such as prefixes, suffixes, and root words) instead of full words, which can improve the system's ability to handle variations in word forms (such as plural vs. singular forms or verb tense changes).

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this study, a joint Transformer and CTC model was employed for Amharic speech recognition systems and verified its effectiveness on Amharic datasets developed by [5]. It is a hybrid sequence-to-sequence model based entirely on self-attention without using RNNs or convolutions. This joint Transformer-CTC model improves the convergence rate of the Transformer model; incorporating both resulted in a faster convergence time than when only the Transformer was used. Furthermore, characters and subwords were compared as the recognition units in joint Transformer and CTC Amharic attention-based models. Our test results showed that using character language units resulted in a lower character error rate, while using subword language units resulted in the best word error rate. However, we urged that since our dataset was so small, increasing the subword units might have a negative impact on accuracy because of the sparse occurrences. Additionally, the study demonstrated the effectiveness of integrating an external LM to enhance recognition accuracy.

Based on the findings, it can be concluded that joining a Transformer and CTC is a promising way to speed up the convergence of the Transformer model. When developing Amharic speech recognition systems, it is crucial to choose the proper language unit (character or subword) depending on the goal: greater word accuracy or lower character error rates. It should be noted that while larger subword units might enhance accuracy, they might also result in sparse distributions of subword units if data is scarce. Moreover, an increase in Amharic speech recognition accuracy is made possible by incorporating language models into the joint Transformer and CTC models.

5.2 Future Work

Although we have made significant progress in building an Amharic speech recognition system based on joint Transformer and CTC, we acknowledge that there is still much work to be done in this area. We recommend looking into the following crucial areas to advance research progresses:

First, it should be mentioned that only a read speech corpus was used in our study. Using only a read speech corpus is not enough to develop an efficient Amharic speech recognition system. Hence, we recommend the development of spontaneous speech corpora to enable the use of the Amharic speech recognition system in practical settings.

Second, while the read speech corpus serves as a crucial foundation, it can be expanded to increase its size, quality, and coverage in order to accelerate the convergence of the Transformer model.

Additionally, we restricted our research to Amharic ASRS using the joint Transformer and CTC approach. Therefore, it will be crucial to investigate and contrast various models in the future in order to gain a thorough understanding of the best strategies for Amharic speech recognition. To ensure that proposed models are tested and compared using the same metrics, datasets, and conditions, robust evaluation practices should be adopted. By putting these suggestions into practice, Amharic speech recognition systems would be significantly improved, and their adoption would be made easier.

Bibliography

[1] Nirayo Gebreegziabher and Andreas Nürnberger. Sub-word based end-to-end speech recognition for an under-resourced language: Amharic. 10 2020.

[2] Solomon Abate, Wolfgang Menzel, and Bairu Tafila. An amharic speech corpus for large vocabulary continuous speech recognition. pages 1601–1604, 09 2005.

[3] Nirayo Gebreegziabher and Andreas Nürnberger. A light-weight convolutional neural network based speech recognition for spoken content retrieval task. 10 2020.

[4] Eshete Emiru, Yaxing Li, Awet Fesseha, and Moussa Diallo. Improving amharic speech recognition system using connectionist temporal classification with attention model and phoneme-based byte-pair-encodings. *Information*, 12:62, 02 2021.

[5] Nirayo Gebreegziabher and Andreas Nürnberger. *An Amharic Syllable-Based Speech Corpus for Continuous Speech Recognition*, pages 177–187. 09 2019.

[6] Geoffrey Hinton, li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Phuongtrang Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29:82–97, 11 2012.

[7] Rohit Prabhavalkar, Kanishka Rao, Tara Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. A comparison of sequence-to-sequence models for speech recognition. pages 939–943, 08 2017.

[8] Yanzhang He, Tara Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, and Alexander Gruenstein. Streaming end-to-end speech recognition for mobile devices. pages 6381–6385, 05 2019.

[9] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8), 2019.

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks. volume 2006, pages 369–376, 01 2006.

[11] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. pages 4960–4964, 03 2016.

[12] Rohit Prabhavalkar, Tara Sainath, Bo Li, Kanishka Rao, and Navdeep Jaitly. An analysis of ”attention” in sequence-to-sequence models. 2017.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] W. Leslau. *Introductory Grammar of Amharic*. Edition Akzente. Harrassowitz, 2000.

[15] Baye Yimam. *y¨aamarIˇna s¨awas¨aw*. 2nd. ed. edition. EMPDE, Addis Ababa, 2007.

[16] Martha Tachbelie and Wolfgang Menzel. Amharic part-of-speech tagger for factored language modeling. pages 428–433, 01 2009.

[17] Nirayo Gebreegziabher and Sebsibe Hailemariam. Modeling improved syllabification algorithm for amharic. 10 2012.

[18] Martha Tachbelie, Solomon Abate, Laurent Besacier, and Solange Rossato. Syllable-based and hybrid acoustic models for amharic speech recognition. 01 2012.

[19] Solomon Abate and Wolfgang Menzel. Syllable-based speech recognition for amharic. pages 33–40, 06 2007.

[20] Martha Yifiru Tachbelie, Solomon Teferra Abate, and Wolfgang Menzel. Morpheme-based automatic speech recognition for a morphologically rich language - amharic. In *Workshop on Spoken Language Technologies for Under-resourced Languages*, 2010.

[21] Martha Tachbelie, Solomon Abate, and Laurent Besacier. Using different acoustic, lexical and language modeling units for asr of an under-resourced language – amharic. *Speech Communication*, 56:181–194, 01 2014.

[22] Adey Edessa and Martha Tachbelie. Investigating the use of syllable acoustic units for amharic speech recognition. pages 1–5, 09 2015.

[23] Tom Bäckström, Okko Räsänen, Abraham Zewoudie, Pablo Pérez Zarazaga, Liisa Koivusalo, Sneha Das, Esteban Gómez Mellado, Mariem Bouafif Mansali, and Daniel Ramos. *Introduction to Speech Processing: 2nd Edition*. Zenodo, July 2022.

[24] solomon Birhanu. Isolated amharic consonant-vowel syllable recognition: Experiment using the hidden markov model. 2001.

[25] Solomon Abate, Wolfgang Menzel, and Bairu Tafila. An amharic speech corpus for large vocabulary continuous speech recognition. pages 1601–1604, 09 2005.

[26] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu. Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese. pages 791–795, 09 2018.

[27] Shigeki Karita, Nelson Yalta, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. pages 1408–1412, 09 2019.

[28] Daniel Jurafsky and James Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2. 02 2008.

[29] Amir Hussein, Shinji Watanabe, and Ahmed Ali. Arabic speech recognition by end-to-end, modular systems and human, 01 2021.

[30] Vassilis Digalakis and Leonardo Neumeyer. Acoustic modelling. pages 240–249, 09 2000.

[31] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu. A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese, 05 2018.

[32] Zhangyu Xiao, Zhijian Ou, Wei Chu, and Hui Lin. Hybrid ctc-attention based end-to-end speech recognition using subword units. 04 2019.

[33] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. pages 4835–4839, 03 2017.

[34] Andargachew Mekonnen, Michael Gasser, Andreas Nürnberger, and Binyam Seyoum. Contemporary amharic corpus: Automatically morpho-syntactically tagged amharic corpus. 10 2018.

[35] Eldan Cohen and Christopher Beck. Empirical analysis of beam search performance degradation in neural sequence models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1290–1299. PMLR, 09–15 Jun 2019.

[36] Clara Meister, Tim Vieira, and Ryan Cotterell. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809, 2020.

[37] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. Espnet: End-to-end speech processing toolkit, 2018.