Amharic Speech Recognition Using Joint Transformer and Connectionist Temporal Classification with Character-Based and Sub-word-Based Acoustic and Language Models

Alemayehu Yilma Demisse[1] and Bisrat Derebssa Dufera[1,*]

[1] School of Electrical and Computer Engineering, Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa, Ethiopia
** Corresponding author's Email address: bisrat@aait.edu.et bisrat@aait.edu.et*

DOI: https://doi.org/10.20372/zede.v42i.10187

ABSTRACT

Sequence-to-sequence attention-based models have gained considerable attention in recent times for automatic speech recognition (ASR). The transformer architecture has been extensively employed for a variety of sequence-to-sequence transformation problems, including machine translation and ASR. This architecture avoids sequential computation that is used in recurrent neural networks and leads to improved iteration rate during the training phase. Connectionist temporal classification, on the other hand, is widely employed to accelerate the convergence of the sequence-to-sequence model by explicitly learning a better alignment between the input speech feature and output label sequences. Amharic language, a Semitic language spoken by 57.5 million people in Ethiopia, is a morphologically rich language that poses a challenge for continuous speech recognition as a root word can be conjugated and inflected into thousands of words to reflect subject, object, tense and quantity. In this research, the connectionist temporal classification is integrated with the transformer for continuous Amharic speech recognition. A suitable acoustic modeling unit for Amharic speech recognition system is also investigated by utilizing character-based and sub word-based models. The results show that a best character error rate of 8.04 % for the character-based model with character-level language model (LM) and a best word error rate of 22.31 % for the sub word-based model with sub word-level LM.

Keywords: Amharic, ASR, CTC**,** LMs, RNNs, Transformer,

1. INTRODUCTION

ASR has a wide range of applications including security, e-health, education, and transport systems, making it an important and active research domain. Research on Amharic ASR has been conducted using various methods. However, the development of ASR for Amharic, like other under-resourced languages, remains a challenging task due to the lack of high-quality language resources. Despite the challenges, ongoing research continues to improve the performance of Amharic ASR systems [1].

In recent years, ASR systems have undergone a significant transition from a hybrid HMM modeling approach [2] to an end-to-end or all neural networks modeling approaches [3, 4]. In contrast to the traditional models, which comprise a number of independent components, the end-to-end structure portrays the system as a single neural network [5].

End-to-end systems are exemplified by models such as the connectionist temporal classification (CTC) [6] and the attention-based encoder-decoder [7]. The CTC based acoustic model (AM) training does not need the frame level alignments between characters in the transcript and the observed input speech [6]. This is due to CTC

introducing a "blank label" which determines the start and end of one character [6]. In the attention-based encoder-decoder models, the encoder is analogous to AM that transforms input speech into higher-level representation, and also to LM that predicts each output token as a function of the prior prediction. The attention mechanism on the other hand is an alignment model to determine frames to predict the next token [8].

Recurrent neural networks (RNNs) are the basis of the end-to-end ASR 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 transformer has been proposed [9]. This architecture eliminates recurrence and relies on its internal attention (self-attention) mechanism without using RNNs 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.

Several research studies [1, 10-12] have been done to develop a continuous speech recognition system for Amharic language using traditional HMM [10-12] and DNN [13-15] approaches. HMM-GMM paradigm with intermediate components [1, 10-12] has been employed to come up with an ASR system for Amharic language. Although they produced relatively acceptable results in the past [15], 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 AMs. In addition, the HMM model for speech has some inherent limitations. HMM is unable to represent contextual information, which could lead to misidentification in long sentences with complex structures. This is because the transitions between each state depend only on the current state and not on any information from previous states. HMMs are based on the assumption that the observations are independent, however speech signals are interdependent and highly non-linear in nature, which means HMM have difficulty in capturing these complex relationships between the observation sequences.

Consequently, few Amharic ASR research have concentrated on end-to-end modeling techniques such as CNN and RNN [14, 15], which seek to instantly simulate the translation between speech and labels without the need of intermediary components. Hybrid CTC and attention model with grapheme to phoneme conversion algorithms was proposed [13] to model sub-word level Amharic language units to address the problem of out-of-vocabulary words. Attention-based models are usually composed of encoder and decoder, which both consist of RNNs. However, in RNNs, the input is reliant on previous time steps, and hence calculations can only be done in sequence.

Transformer and RNN based ASR were combined by Syoun et. al. [16] to develop a faster and more accurate ASR system. A CTC with transformer is utilized for co-learning and decoding to develop the model. This strategy expedites learning and assists with LM integration. Significant advancements in many ASR tasks are implemented by the suggested 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 % while integrating CTC and LM into the transformer baseline. Transformer based paradigm for online streaming ASR that needs a

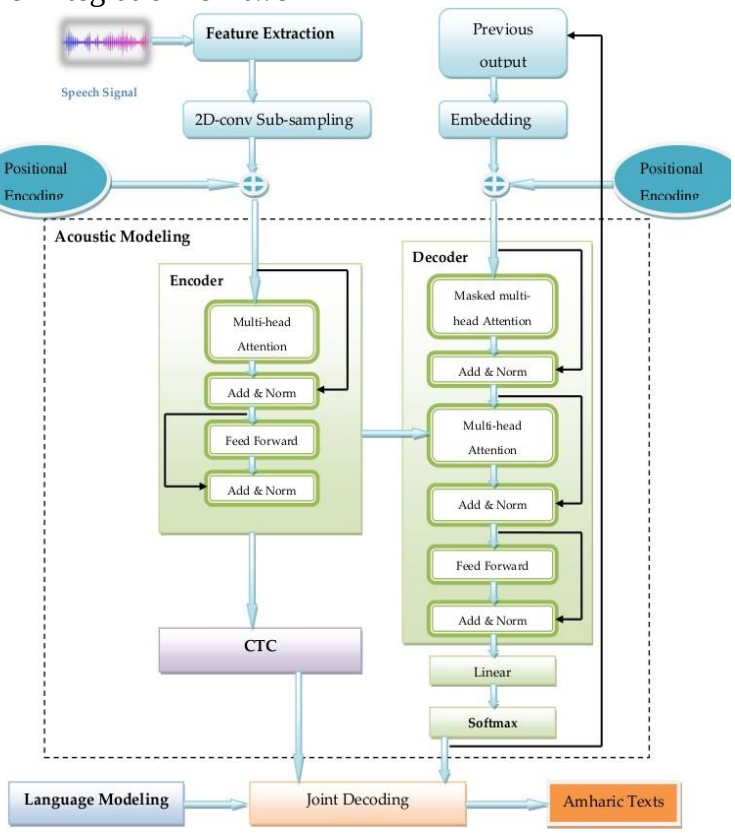
continuous speech as input was presented in a study [17]. In this work, an output is generated promptly after each utterance. They employed time-restricted self-attention for the encoder and triggered attention for the encoder-decoder attention mechanism. Their model resulted of WER 2.8 % and 7.3 % for the "clean" and "other" Libri Speech test data, respectively.

In this paper, we propose joint CTC and attention-based model with transformer architecture for Amharic continuous speech recognition. This architecture removes recurrence and relies on self-attention mechanism to determine relationships between input and output, which allows for parallelization. This research presents two significant contributions that aim to improve the accuracy of ASR for Amharic language. Firstly, it proposes the integration of two cutting-edge ASR techniques, namely CTC and transformer joint training, which enables modeling of different Amharic language units (characters and subwords) to achieve better ASR accuracy. Secondly, this research evaluates and analyzes the performance of various Amharic language modeling units, including character RNNLM and subword RNNLM.

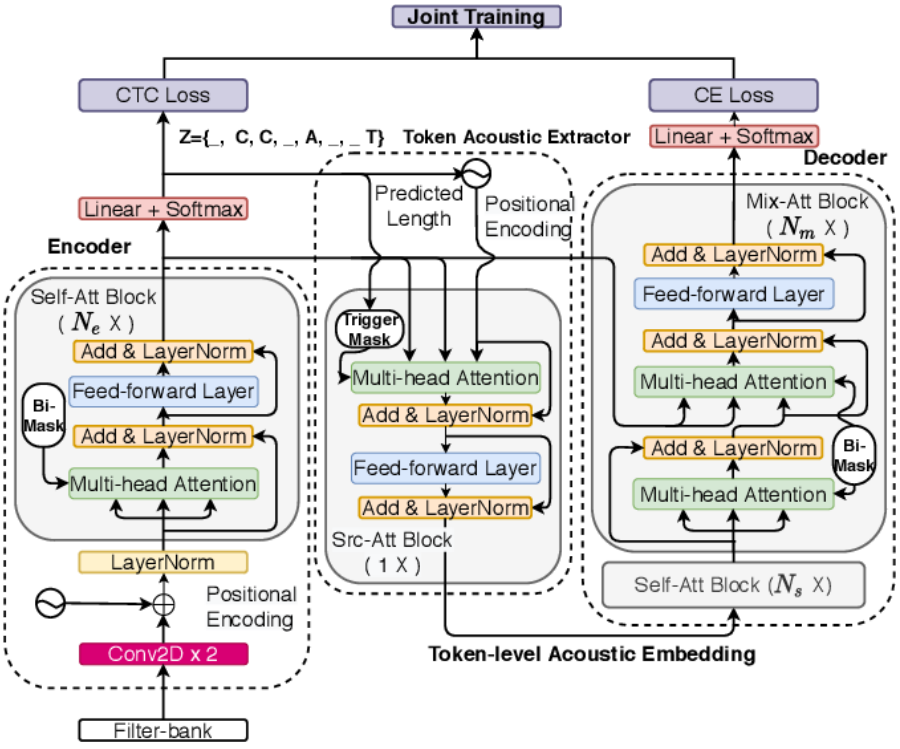
As far as we can tell from our reading, no work has been published that employs the transformer-based end-to-end architecture for Amharic ASR tasks.

2. METHODS

The proposed model shown in Figure 1 consists of five pivotal stages, namely feature extraction, sub-sampling, AM, LM and joint decoding.



(a)



(b)

Figure 1 (a) The proposed model architecture for Amharic ASR and (b) CTC architecture.

2.1. Feature extraction

Feature extraction is a critical step in the process of ASR, as it helps to transform the raw audio data into manageable, relevant, and informative features. In this study, the log-Mel filter bank features are utilized to provide a compact representation of the input signal by computing a series of feature vectors.

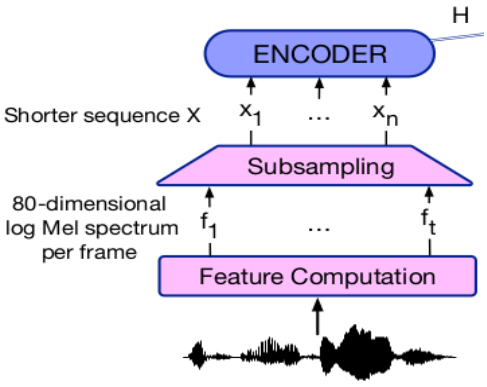


Figure 2 Schematic architecture showing pre-encoder stages.

2.2. Sub sampling

The encoder-decoder architecture ideally suits scenarios where input and output sequences have similar lengths. A single word could consist of five letters and go on for around 2 seconds, equating to approximately 200 acoustic frames (at 10ms per frame). Due to this significant length disparity, speech-based encoder-decoder architectures require to employ a compression stage. This stage pre-processes the speech features, typically shortening their sequence length before feeding them into the encoder.

One popular method is using 2D-conv subsampling [5], which involves reducing the size of the input while preserving essential features. After subsampling, the feature frames F are transformed into sub-sampled sequence $X \in R^{d^{sub} x\, d^{model}}$ with 2D-CNN sampling layer as shown in Figure 2.

2.3. Acoustic Modeling

In speech recognition, acoustic modeling involves creating statistical models that represent how sounds in speech relate to linguistic units. The joint transformer-CTC model is considered as an AM. This framework utilizes a shared transformer encoder to generate a high-level representation $h = (h_1,h_2,...,h_L)$ 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 representation h.

2.3.1. Transformer Architecture

Transformer uses sinusoidal position information and a self-attention mechanism to completely do away with repetitions in typical RNNs [18, 19]. It is made-up of one large block, which itself is made up of blocks of encoders and decoders.

The encoder's core function is to transform the input sequence into a high-level representation using a combination of two techniques: multi-head self-attention and a fully-connected network with positional encoding. 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^v, W^q and $W^k \in R^{d^{model}\times d^k}$, where d^{model} represents the dimension of the output of the previous attention layer. Moreover, $d^q = d^k$, d^v, symbolize the dimensions of queries, keys and values, respectively. A normalized weighted similarity Z is obtained from self-attention using softmax, which is shown in Eq. (1).

$$\text{SelfAttention}(Q, K, V) = \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d^k}}\right) \times V \quad (1)$$

Multi-head attention (MHA) tackles the challenge of attending to different aspects of the input simultaneously. It achieves this by applying multiple, parallel attention sub-layers, each focusing on different features or relationships within the data. MHA comprises concatenating all self-attention heads at a specific layer (see Eqs (2) and (3)).

$$MHA(Q,K,V)=[Z_1,Z_2,...,Z_h]\ W^h \quad (2)$$

$$Z_i=SelfAttention(Q_i,K_i,V_i) \quad (3)$$

After passing through the multi-head attention layer, the resulting representation is normalized and fed into a fully-connected neural network layer known as the feed-forward sub-layer, Eq. (4).

$$FF(z[t])=max(0,z[t]\times W_1+b_1)W_2+b_2 \quad (4)$$

where:

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

The decoder generates predictions in an auto-regressive manner. At each time step, it utilizes the high-level representation from the encoder and previous predictions from the decoder as inputs for the current prediction. At each time step, the decoder generates a prediction $\hat{Y}[t]$that hinges on the final encoder representation H_e andthe prior target sequence $Y[1: t-1]$. To achieve such conditional dependence, the decoder deploys multi head attention, enabling it to calculate attention between encoder high-level features and previously decoded sequences. Similar to the encoder, the decoder comes complete with layer normalizations and residual connections focused around every sub-layer.

The transformer employs two fundamental mechanisms, namely Positional Encoding (PE) and Embeddings. These crucial techniques are responsible for encoding the

positional information of the input sequence and learning representations for each token, respectively. 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. It provides valuable indication regarding the order of the words in the sequence to the model. On the other hand, Embeddings are a way to represent each token as a dense vector. 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 for the model to learn semantic relationships between the tokens, allowing it to generalize better by understanding the context of each token in the sequence.

2.3.2 Connectionist Temporal Classification

CTC: is a novel method that has revolutionized the way in which transformers are trained. CTC leverages a unique approach that does not require any previous alignment among input and output sequences of varying lengths [28]. Instead, it presents a high-level variable, known as the CTC path $\pi = (\pi_1, \pi_2,...,\pi_L)$ for the input sequence as a frame-level label.

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. Once the transformer encoder processes the input, it generates a high-level representation capturing the essential information. This representation is then utilized for subsequent processing stages in the speech recognition system [6]. The probability of a CTC path can be computed by using Eq. (5).

$$p\left(\frac{\pi}{x}\right) = \prod_{l=1}^{L} q_l^{\pi_l} \tag{5}$$

The likelihood of the label sequence is the sum of probabilities of all compatible CTC paths (see Eq. (6)).

$$p\left(\frac{y}{x}\right) = \sum_{\pi \in \Phi(y)} p\left(\frac{\pi}{x}\right) \tag{6}$$

where:

$\Phi(y)$ denotes the set of all 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^u and the backward variable β^u as shown in Eq. (7).

$$p\left(\frac{y}{x}\right) = \sum_{u} \frac{\alpha_l^u \beta_l^u}{q_l^{\pi_l}} \tag{7}$$

where:

u is the label index.

The CTC loss is defined as the negative log likelihood of the output label sequence, (Eq. (8)).

$$L_{CTC} = -\ln\left(p\left(\frac{y}{x}\right)\right) \tag{8}$$

The CTC loss can be used to train the transformer Encoder by using the back-propagation algorithm by derivation of the CTC loss.

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, transformer attention mechanism uses a weighted sum over all inputs without constraints from alignments, resulting in difficulties when training the transformer-

based decoder.

The joint CTC-transformer objective function is the weighted sum of the transformer loss and CTC loss (Eq. (9)).

$$L_{joint} = \lambda L_{CTC} + (1-\lambda) L_{Transformer} \quad (9)$$

where:

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

2.4. Language Modeling

Language modeling refers to the process of predicting the likelihood of a sequence of tokens in a given language. Given that a transformer model is fundamentally a conditional LM, it implicitly learns a LM for the intended output domain via its training data.

Character level and subword level LMs for Amharic speech recognition were developed using LSTM. LSTMs are a type of re-current neural network that can learn long-term relationships in sequential data. They are particularly useful for language modeling because they can capture the context of a token and its impact on the following tokens in a sentence. In language modeling, the model receives a sequence of words or symbols as input, and predicts the likelihood of the next one in the sequence. The LSTM takes one token at a time as the input and based on previous state and current token it updates its internal state. The hidden state of the LSTM effectively captures the context of the sentence up to that point, allowing the model to predict the most likely next token.

2.1.5 Joint Decoding

In the decoding process, a LM is employed to distinguish and clarify between the expected sentences that are produced by the transformer decoder. By utilizing beam search, we obtain the final selection of hypothesized sentences in the form of an n-best list. These hypotheses are then rescored using a LM, whereby each hypothesis scored on the beam is recalculated. As can be seen in Eq. (10), this score is calculated by joining the score obtained from the LM with the CTC score.

$$\hat{y} = argmax(\lambda \log \rho_{s2s}\left(\frac{y}{x}\right) + (1-\lambda) \log \rho_{ctc}\left(\frac{y}{x}\right) + \gamma \log \rho_{lm}(y) \quad (10)$$

where:

$\rho_{s2s}\left(\frac{y}{x}\right)$ is the transformer decoder probability of the output sequence given theencoding feature sequence,

$\rho_{ctc}\left(\frac{y}{x}\right)$ 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 hyper parameters named "CTC weight" and "LM weight", respectively.

2.2 EXPERIMENTAL EVALUATION

2.2.1 Dataset

In this study, the Amharic speech corpus prepared Solomon Abate et al. [21], which comprises approximately 110hours of speech obtained from 214 speakers (male and female in equal proportion) who read a total of 32,901 sentences were used. The sentences were obtained from the archive of Ethio Zena website which focuses on news related sentences. The dataset was split into training, validation and test set, which contains 29, 221 sentences, 500 sentences and 3180 sentences, respectively. All the dataset has both character-based and syllable-based transcription for each utterance.

2.2.2 ASR Evaluation Metrics

Character error rate (CER) and word error rate (WER) were taken as evaluation metrics of the proposed method.

CER is a metric used to assess the performance of systems that deal with text, like ASR and Optical Character Recognition

(OCR). It is the percentage of characters that were incorrectly processed by the system. A lower CER indicates better performance, with 0% being a perfect score.CER is useful because it focuses on individual characters, providing a more granular view of errors compared to metrics that look at entire words. This can be helpful in identifying specific issues with pronunciation or recognition. CER was evaluated using Equation 11.

$$CER = (S + D + I) / N \quad (11)$$

where:

CER is Character Error Rate (percentage)
S is Number of substitutions (incorrect characters)
D is Number of deletions (missing characters)
I is Number of insertions (extra characters)
N is Total number of characters in the reference text (ground truth)

WER were another common metric used to evaluate the performance of speech recognition and machine translation systems. It focuses on errors at the word level, rather than individual characters like CER. It is defined similar to CER except word is used instead of character.

2.2.3 Experiment Setup

The training and testing experiments were conducted using Google Colab, a convenient cloud-based service courtesy of Google. The transformer model consists of twelve encoder layers and six decoder layers that form a 2048-dimensional feed-forward network. Eight attention heads were used, each with 512 dimensions, to provide our system with increased attentiveness.

To implement the joint training method, a multi-task loss weight of 0.3 for CTC was used. To avoid risk of over fitting several regularization techniques were incorporated, including 10% dropout on every attention matrix and weight in feed forward (FF), layer normalization before every MHA and FF, as well as a penalty of 0.1 as label smoothing, effectively preventing over fitting. Training was conducted with over 100 epochs using Pytorch modeling and a batch size of 8. Further, the Noam optimizer with warm up steps, label smoothing, gradient clipping, and accumulating gradients were utilized to train the proposed speech recognition system.

A sampling rate of 16 kHz, audio frames of 25ms duration intervals separated by an interval of 10ms, leading to the extraction of 80-dimensional log mel-filter bank features, were used for both training and decoding.

Two distinct types of LMs were explored, namely sub-word units, and character units. In order to achieve optimal results with sub-word LM, a 2-layer LSTM architecture that consisted of 1024 hidden units supplemented with Noam optimization, a batch size of 64, and a maximum sequencelengthof55 was used. The sub word LM was found to be particularly useful for modeling the complex structure of words and phrases that do not appear in their entirety in the training data. Alternatively, character LM employed a 4-layered LSTM architecture, with each layer containing 512 hidden units. Similar to the sub-word model, the character LM also utilized Noam optimization to enhance performance. This model's batch size was increased to 256 batch size because character-level modeling often has longer sequences. The maximum sequence length was set at 400 characters. The character LM is particularly valuable when focusing on morphology or spelling patterns across diverse languages.

3. RESULTS AND DISCUSSIONS

Training on both transformer model and joint transformer-CTC model is shown in Figure 3

for character-level tokenization. The results in Figure 3 shows that the transformer model failed to converge even after increasing the number of epochs. On the other hand, the transformer-CTC joint training achieved faster convergence. The reason for such significant differences between the two models 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. This in turn, allowed the framework to converge much more effectively and efficiently. These results not only provide valuable insights into the limitations of transformer models but also highlight the importance of using CTC joint training in ASR tasks.

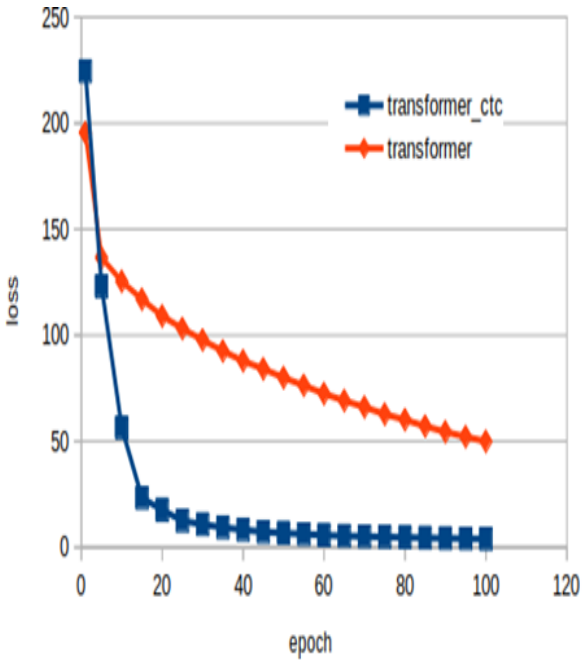


Figure 3 Training losses in character-based recognition

3.1 LM Training Results

As can be seen in Figure 4 and Figure 5 the perplexity is 6.35 and 10.24 for character-level and subword-level LMs, respectively for the validation dataset. The decrease in validation perplexity over epochs suggests that both models did not over fit on the training data and thus should generalize better on unseen test data.

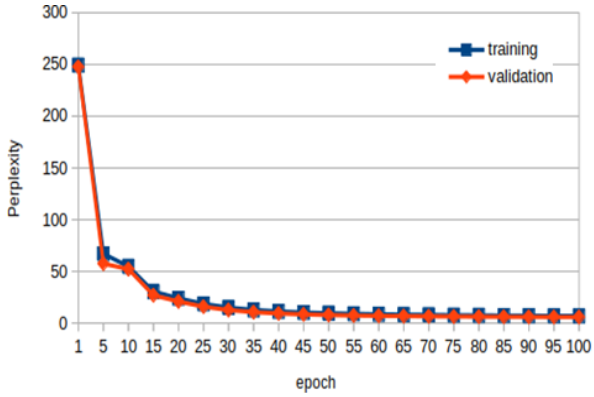


Figure 4 Training and validation perplexities of character-level LM

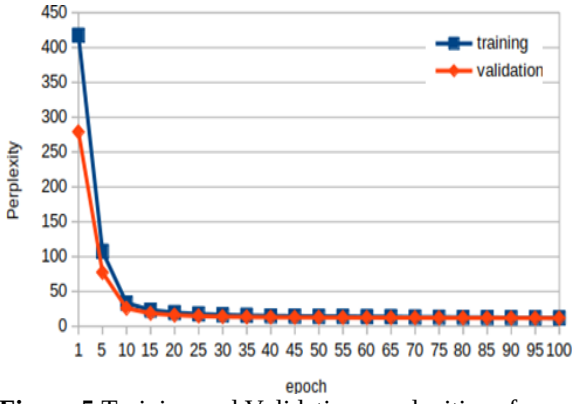


Figure 5 Training and Validation perplexities of subword-level LM

3.2 Joint Decoding

The results of the proposed joint transformer-CTC decoding have been categorized into two main categories: character-based AM with both character-level and subword-level L M, and subword-based AM with both character-level and subword-level LM. The models have been evaluated for their ability to decode an unseen test data, with the ultimate goal of achieving maximum transcription accuracy.

1) Character-based AM:

Table 1 depicts the performance of the character-based AM model. The initial results reveal a CER of 8.53% and a WER of 26.39 %, without the incorporation of LM.In order to enhance the performance of the model, distinct LMs such as a character-level

LM and a sub word-level LM were integrated into the decoding process.

Upon incorporating the character-level LM, both the CER and WER improved to 8.04% and 24.71% respectively. In contrast, the use of the sub-word-level LM resulted in a higher CER of 8.89% but a lower WER of 23.56%. This observation shows that even though character-level LM improves character recognition, in terms of word recognition sub-word-level LM is superior. The difference in performance between the character-level and subword-level LMs can be attributed to their inherent characteristics. Specifically, a sub-word-level LM is better suited to capturing the most probable sequence of characters that makeup a word, which can improve overall WER.

2) Sub-word-based AM:

Table 2 illustrates the performance subword-based joint transformer and model on unseen test data. The first result of the study showed that a model with 600 subword units achieved a CER of 9.21 % and a WER of 25.07 %. When a model with 2000 subword units is used, the CER increased significantly to 23.42 %, and the WER also increased to 42.3 %. This suggested that the high level of complexity resulting from more subword units creates over fitting. Additionally, increasing the number of subword units leads to sparser unit occurrences in the text corpus.

Table 1 Decoding results of joint transformer and CTC model using character as recognition unit

LM	CER		WER	
	Greedy decoding	Beam search decoding (width=3)	Greedy decoding	Beam search decoding (width=3)
No LM	9.43 %	8.53 %	28.2 %	26.39 %
Character -level	-	8.04 %	-	24.71 %
Subword -level	-	8.89 %	-	23.56 %

Table 2 Decoding results of joint transformer and CTC model using subword as recognition unit

Number of subwords	LM	CER		WER	
		Greedy decoding	Beam search decoding (width=3)	Greedy decoding	Beam search decoding (width=3)
600	No LM	10.71 %	9.21 %	26.53 %	25.07 %
2000	No LM	27.21 %	23.42 %	46.6 %	42.3 %
600	Character-level	-	8.85 %	-	24.02 %
600	Subword -level	-	9.20 %	-	22.31 %

To improve the model's performance, character-level and subword-level LMs were incorporated. The results demonstrated a significant improvement in accuracy when the character-level LM was utilized, with CER reduced to 8.85 % and WER to 24.02 % when using 600 subword units. On the other hand, incorporating a subword-level LM achieving a CER of 9.2 % and a WER of 22.31 %. Even though the character-level LM improved the character recognition, the subword-level LM achieved better WER as it captured the complex patterns of subwords present in the text and provided context to the model for more accurate predictions. This demonstrated that subword-based joint transformer and CTC models do not necessarily perform better with increasing subword units. Therefore, an optimal number of subword units is crucial to avoid having a model that is overly complex, leading to over fitting, as previously discussed. It is also worth noting that in this application beam search decoding outperformed greedy

decoding in all experiments, indicating that it is superior in decoding sequence of speech recognition hypothesis

4. CONCLUSIONS

In this study, we sought to investigate joint transformer and CTC in enhancing the accuracy of speech recognition. To assess the effectiveness of this approach, we employed two decoding techniques: greedy decoding and beam search decoding. Our findings revealed that beam search decoding outperformed greedy decoding in all experiments, indicating that it is superior in decoding sequences of speech recognition hypotheses. Specifically, our results demonstrate that beam search is a more viable strategy than greedy decoding for improving upon speech recognition accuracy utilizing joint transformer and CTC. The results of the evaluation demonstrated that subword-based models perform better than character-based models. Based on the findings, it can be concluded that using a joint transformer and CTC represents a promising method to achieve faster convergence of the transformer model. Additionally, selecting the appropriate language unit (character or subword) when developing Amharic speech recognition systems, is crucial and dependent on the end objective: higher word accuracy or lower character error rates.

CONFLICTS OF INTEREST

The authors have no conflict of interests related to this publication.

ACKNOWLEDGMENTS

The researchers would like to acknowledge School of Electrical and Computer Engineering, Addis Ababa Institute of Technology, Addis Ababa University for providing the computational facility to undertake this research.

REFERENCES

[1] Eshete Emiru, Yaxing, L., Awet Fesseha, and Moussa, D. "*Improving Amharic speech recognition system using connectionist temporal classification with attention model and phoneme-based byte-pair-encodings*", Information, vol. 12, February 2021. doi: https://doi.org/10.3390/info12020062.

[2] Amir, H., Shinji, W., and Ahmed, A. "*Arabic speech recognition by end-to-end, modular systems and human, Computer Speech & Language*", Computer Speech & Language, vol. 71, 2022.

[3] Geoffrey, H., Li,D., Dong, Y., George, D., Abdelrahman, M., Navdeep, J., Andrew, S., Vincent, V, Phuongtrang, N., Tara, S., and Brian, K."*Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*", Signal Processing Magazine, IEEE, vol. 29, pp. 82–97, November 2012, doi:10.1109/MSP.2012.2205597.

[4] Rohit, P., Tara, S., Bo, L., Kanishka, R., and Navdeep, J., "*Analysis of "attention" in sequence-to-sequence models*" In proceeding of 18th Annual Conference of the International Speech Communication Association (Interspeech), Stockholm, Sweden, August 20-24, 2017

[5] Dong, W., Xiaodong, W., and Shaohe, L. "*An overview of end-to-end automatic speech recognition*". Symmetry, vol. 11, no. 8, 2019. doi: 10. 3390/sym11081018.

[6] Yanzhang, H., Tara, S., Rohit, P., Ian, M., et. al., "*Streaming end-to-end speech recognition for mobile devices*", International Conference on Acoustics, Speech, & Signal Processing (ICASSP), 12-17 May, 2019, Brighton, England, doi: 10. 1109/ICASSP.

[7] Chan, W., Jaitly, N., Le, Q. and Vinyals, O., *"Listen, attend and spell: A neural network for large vocabulary conversational speech recognition*", 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016, pp. 4960-4964, doi: 10.1109/ICASSP.2016.7472621.

[8] Martha Yifru Tachbelie, Solomon Abate, and Laurent, B., *"Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic"*, Speech Communication, vol. 56 pp. 181–194, 2014, doi:10.1016/j.specom.2013.01.008

[9] Fergus, R., Vishwanathan, S., and Garnett, R., "*Advances in Neural Information Processing System,"*, MIT Press, volume 30. Curran Associates, Inc., 2017.

[10] Solomon Abate and Wolfgang, M., "*Syllable-based speech recognition for Amharic"*, In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, June 2007. doi:10.3115/1654576.1654583.

[11] Martha Yifiru Tachbelie, Solomon Teferra Abate, and Wolfgang, M., "*Morpheme-based automatic speech recognition for a morphologically rich language–Amharic",* In Workshop on Spoken Language Technologies for Under-resourced Languages, Penang, Malaysia, 2010.

[12] Ashish, V., Noam, S., Niki, P., Jakob, U., Llion, J., Aidan, N. Go., Łukasz, K., and Illia, P., "*All you need is attention",* Neural Information Processing Systems, 2017

[13] Nirayo Gebreegziabher N. and Sebsibe Hailemariam, "*Modeling improved syllabification algorithm for Amharic"*, In Proceedings of the International Conference on Management of Emergent Digital Eco Systems (MEDES '12). Association for Computing Machinery, New York, NY, USA, 16–21, 2012.

[14] Gebreegziabher, N., and Nürnberger, A., "*Sub-word Based End-to-End Speech Recognition for an Under-Resourced Language: Amharic*", 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 3466-3470, doi:10.1109/SMC42975.2020.9283401.

[15] Alex, G., Santiago, F., Faustino, G., and Jürgen, S., "*Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks*", In Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery, New York, NY, USA, 369–376. doi.org/10.1145/1143844.1143891.

[16] Suyoun, K., Takaaki, H., and Shinji, W., "*Joint ctc-attention based end- to-end speech recognition using multi-task learning",* ICASSP-2017, pages4835–4839, 03 2017. doi:10.1109/ICASSP.2017.7953075.

[17] Rohit, P., Kanishka, R., Tara, S., Bo,L., Leif, J., and Navdeep, J., "*A comparison of sequence-to-sequence models for speech recognition",* Interspeech, 2017, pages 939–943

[18] Zhou, S., Dong, L., Xu, S., & Xu, B.,"*A Comparison of Modeling Units in Sequence-to-Sequence Speech Recognition with the Transformer on Mandarin Chinese"*, in proceeding of International Conference on Neural Information Processing, 13-16 December, 2018 Siem Reap, Cambodia ArXiv, abs/1805.06239.

[19] Shiyu, Z., Linhao, D., Shuang, X., and Bo, X., "*Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin Chinese",* Interspeech 2018, September 2-6, Hyderabad, India, pp. 791–795, 2018.

[20] Solomon Abate, Wolfgang M., and Bairu, T., "An Amharic speech corpus for large vocabulary continuous speech recognition", Interspeech-2005, 4-8 September 2005, Lisbon, Portugal, pp. 1601–1604, doi:10.21437/Interspeech.2005-467.