**Edge-Based Speech Transcription and Synthesis for Kinyarwanda and Swahili Languages**

Pacome Simon Mbonimpa, Diane Tuyizere and Natasha Mutangana,

*Carnegie Mellon University Africa*

**Introduction**

In today’s digital age, the need for accurate and efficient speech transcription and synthesis models has been increasing rapidly. These models play an important role in a variety of applications, such as learning a new language(s), accessibility tools for people with difficulties in reading and hearing, as well as automated voice assistants [1]. Kinyarwanda and Swahili are some of the local languages spoken in East Africa. With Swahili being the most widely spoken language, the Swahili speakers vary widely, from 60 million to over 150 million [2]. Swahili serves as the national language of four nations: Tanzania, Kenya, Uganda, Rwanda, and the Democratic Republic of the Congo. On the other hand, Kinyarwanda is the national language of Rwanda, spoken by approximately 24 million people in Rwanda and beyond [3].

Rwanda and Kenya are among the most developing countries in the EAC; although this is the case, there is a significant scarcity of speech transcription and synthesis models for such languages, mainly due to a lack of strong community-based research initiatives to tackle existing computing-related challenges [4]. In addition to that, there are huge, growing technology-backed services in which the availability of speech transcription and synthesis models would not only boost innovation but also contribute to research. Furthermore, the fact that many people in such countries have very little access to the internet as well as possess devices with constrained computation. Having models designed to run seamlessly on such devices and perform locally to the optimum while only requesting further processing to the cloud when the local computation does not meet the accurate threshold would improve the computing efficiency while keeping the accuracy moderate.

In the past, different research efforts have been made to develop Edge-Based speech transcription and synthesis models for languages, including under-resourced ones. For instance, EfficientSpeech [16] provides an On-Device text-to-speech model designed specifically for edge devices. This model aims to provide natural-sounding speech synthesis while being efficient enough to run on-device. To name a few, other approaches such as Tacotron [17], Tacotron 2 [18], and WaveGlow [19], have pushed the boundaries of end-to-end generative text-to-speech models.

On the other hand, approaches that provide edge-based text transcription (speech-to-text) include studies like "Streaming End-to-End Speech Recognition for Mobile Devices" [20] and "Speech Recognition and Speech Synthesis Models for Micro Devices" [21] and so on, which try to integrate the speech recognition models at the edge. Although this is the case, there is still a huge gap in the efficiency and computation-aware intelligence of the deployed architecture models, taking into consideration the computation capacity of the edge device as well as the efficiency in relying on the cloud when support is needed. In addition to that, none of the presented approaches tackle specifically the EAC languages, such as Kinyarwanda and Kiswahili.

In this paper, we present a novel computing-aware edge-based speech transcription and synthesis for Kinyarwanda and Swahili languages. Inspired by the approach proposed in [25], our proposed approach uses a cascading mechanism to efficiently couple the computation between models deployed at the edge as well as the cloud. In a nutshell, a cascading network is capable of distributing a deep neural network between a local device and the cloud while keeping the required communication network traffic to a minimum. The network begins processing on the constrained device and only relies on the remote part when the local part does not provide an accurate enough result. For instance, in TTS, when the processed output voice after the edge processing is very noisy, it has to send the internal representation to the cloud. Similarly, in STT, when the input voice at the edge is noisy, its internal representations are sent to the cloud for better processing. In this report, we presented the results of encoder-decoder models for both speech-to-text as well as text-to-speech for both Kinyarwanda and Swahili.

Consequently, we would summarize the primary contribution of this research paper as follows:

1. Developed speech transcription and synthesis model architectures targeting Kinyarwanda and Swahili languages.
2. Introduced an edge-based system that leverages the Cascading Network architecture to process speech data on local devices efficiently and escalate to cloud-based resources when necessary.
3. Practical deployment of this model within an edge device, incorporating speech-to-text and text-to-speech functionality for enhanced accessibility and usability.
4. Fine-tuned some of the available pre-trained models and adapted them to both Swahili and Kinyarwanda languages.

The remainder of the paper is organized as follows. Section II examines the related work. Section III gives an overview of the approach used, focusing on the architecture of the models considered. Section IV presents the technical results of the experiments and a qualitative discussion of the findings, while Section V concludes the paper and identifies future work.

**II Related Work**

Recent advancements in speech-to-text (STT) and text-to-speech (TTS) technologies begins with an acknowledgment of the significant progress made in machine learning and deep learning architectures. It emphasizes the emerging challenges and opportunities in deploying these advanced models on edge devices, particularly for under-resourced languages like Kinyarwanda and Swahili. This section presents different approaches that cover speech-to-text and text-to-speech technologies while targeting edge device computation.

**1. Speech-to-text models**

In recent years, speech-to-text (STT) models have made significant strides, thanks to advancements in deep neural networks. These models aim to transcribe spoken language into written text, with applications ranging from transcription services to voice assistants.

The survey conducted in [10] provided a synopsis of speech-to-text models that incorporated deep neural networks. Among the various architectures present today, covering various aspects of this task including modeling, training, encoding, and decoding, transformer-based models, such as those presented in [11, 12, 13, 14], have emerged as state-of-the-art, achieving impressive word error rates (WER) on challenging datasets like Librispeech. They reduced the word error rate trend to 3.7% - 1.8% in transcribing the Librispeech dataset.

The variants of the transformer architecture include the "Conformer," a transformer architecture that uses convolutional neural networks to model local dependencies in speech data for transcription [11], [12]. Shifting the task from supervised learning to self-supervised learning and unsupervised learning, the work done in [13] and [14] showed that through pseudo-labeling further improvements, such as a 13% relative WER reduction on more challenging datasets, could be achieved with considerations that supervised learning’s drawbacks include that labeling requires the availability of exhaustive word dictionaries.

Deployment of the models mentioned above critically requires compute capability considerations for the platform. In [11], through varying the number of parameters, three types of the conformer model were proposed: small (10.3 million), medium (30.7 million), and large (118.8 million). With such deep neural network models, tradeoffs between model compression and accuracy need to be examined, per the specifications of the targeted edge device(s). However, network topologies that allow distributing the model across more than a single device would enable workload sharing and maintain accuracy while meeting edge devices’ constraints.

**2. Text-to-speech models**

Text-to-speech (TTS) systems have witnessed significant advancements in recent years, driven by the continuous development of deep learning architectures and training methodologies. These advancements have enabled the generation of increasingly natural-sounding synthetic speech, blurring the lines between human and machine-generated voices. However, a crucial challenge remains in balancing the pursuit of high-fidelity speech synthesis with the need for efficient and resource-constrained models suitable for on-device applications. This section delves into the latest research trends in TTS, exploring the state-of-the-art approaches that address this critical trade-off.

One approach to achieving high-fidelity speech synthesis involves leveraging complex neural network architectures with a focus on maximizing audio quality. One of the current state-of-the-art approaches is FastSpeech 2 [15]. FastSpeech 2 is an innovative end-to-end text-to-speech synthesis model that focuses on generating non-autoregressive mel-spectrograms directly from text. This model represents a significant advancement in speech synthesis technology by enabling the parallel generation of speech waveforms from text, leading to faster inference times compared to its predecessor, FastSpeech. FastSpeech 2 has been recognized for its ability to produce high-quality speech output efficiently. It has also been noted for its superior voice quality compared to other models and its capacity to maintain the advantages of speed, robustness, and controllability in speech synthesis. Additionally, FastSpeech 2 has demonstrated success in achieving fast speech synthesis systems, showcasing its effectiveness in the realm of text-to-speech technology. Despite its remarkable capabilities in generating natural-sounding speech, its substantial computational cost and memory footprint limit its deployment on devices with limited resources.

Recognizing the limitations of high-resource models, researchers are actively exploring alternative strategies for building efficient and resource-constrained TTS systems. EfficientSpeech [16] exemplifies this effort. In the paper "EfficientSpeech: An On-Device Text-to-Speech Model," Atienza introduces a novel text-to-speech (TTS) model designed specifically for edge devices. This model aims to provide natural-sounding speech synthesis while being efficient enough to run on-device. The significance of this work lies in addressing the need for TTS systems that can operate directly on edge devices without relying on cloud services, thus enabling real-time speech synthesis in various applications. EfficientSpeech joins a sequence of advancements in TTS technology, such as Tacotron [17], Tacotron 2 [18], and WaveGlow [19], which have pushed the boundaries of end-to-end generative text-to-speech models.

These models have simplified the traditional speech synthesis pipeline by utilizing neural networks to directly generate speech from text, eliminating the need for intermediate linguistic and acoustic features. The development of EfficientSpeech aligns with the trend towards on-device processing highlighted in studies like "Streaming End-to-End Speech Recognition for Mobile Devices" [20] and "Speech Recognition and Speech Synthesis Models for Micro Devices" [21]. These works emphasize the importance of deploying speech recognition and synthesis models on resource-constrained devices like mobile phones and microcontrollers, enabling applications that require low latency and privacy preservation. LightGrad [22] takes this concept a step further by adopting a non-autoregressive approach that utilizes a lightweight U-Net architecture and streaming inference to achieve even lower latency.

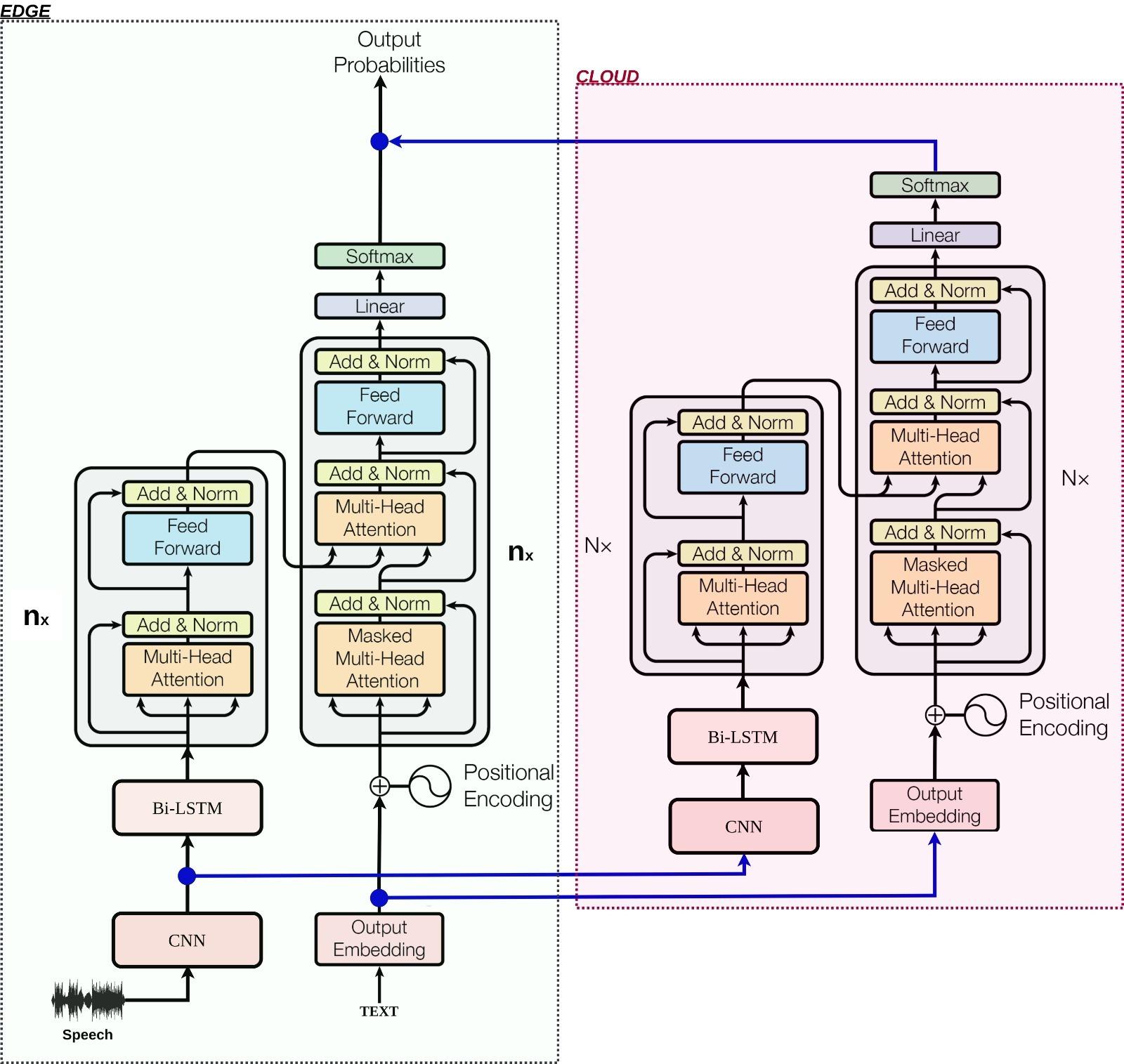
TTS technology is not limited to well-resourced languages. LRSpeech [23] addresses the challenge of building TTS systems for languages with limited data availability. This model leverages pre-training on rich-resource languages, multi-task learning, and knowledge distillation to achieve high speech quality and recognition accuracy even with minimal training data. This opens doors to broader language coverage and caters to the needs of diverse communities, especially those with extremely low-resource languages.

**Methodology**

Our approach for both Speech-To-Text and Text-To-Speech tasks involved the use of cascaded models. These models consist of two interconnected components: an edge encoder-decoder structure and a cloud encoder-decoder structure. The edge encoder processes incoming data (audio signals for Speech-To-Text and texts for Text-To-Speech) and extracts relevant features locally. It is optimized for low-latency inference and operates efficiently on resource-constrained devices. The cloud encoder, situated remotely, complements the edge encoder by providing a broader context. When necessary, it receives feature embeddings from the edge encoder. Similarly, the cloud decoder enhances the edge decoder’s output by generating richer linguistic representations from the embeddings produced at the edge. The final output of the decoder in the cloud is then sent back to the edge device.

1. **Speech-To-Text**

In the Speech-To-Text model, the edge encoder analyzes audio frames and generates embeddings using a combination of a convolutional layer and a Bi-directional LSTM. These embeddings then undergo further processing through a stack of N blocks, each containing a multi-head attention layer followed by a feed-forward network. The decoder component, responsible for text modeling, consists of a text embedding layer followed by a masked multi-head attention mechanism. Additionally, a cross-attention layer incorporates features extracted from the input audio data, leading to text prediction via a SoftMax layer. When faced with challenging acoustic conditions that reduce its average classification confidence below a predefined threshold, the edge device selectively forwards features to the cloud model. The cloud model, with its deeper network architecture for processing embeddings, refines these features, resulting in more accurate transcriptions.

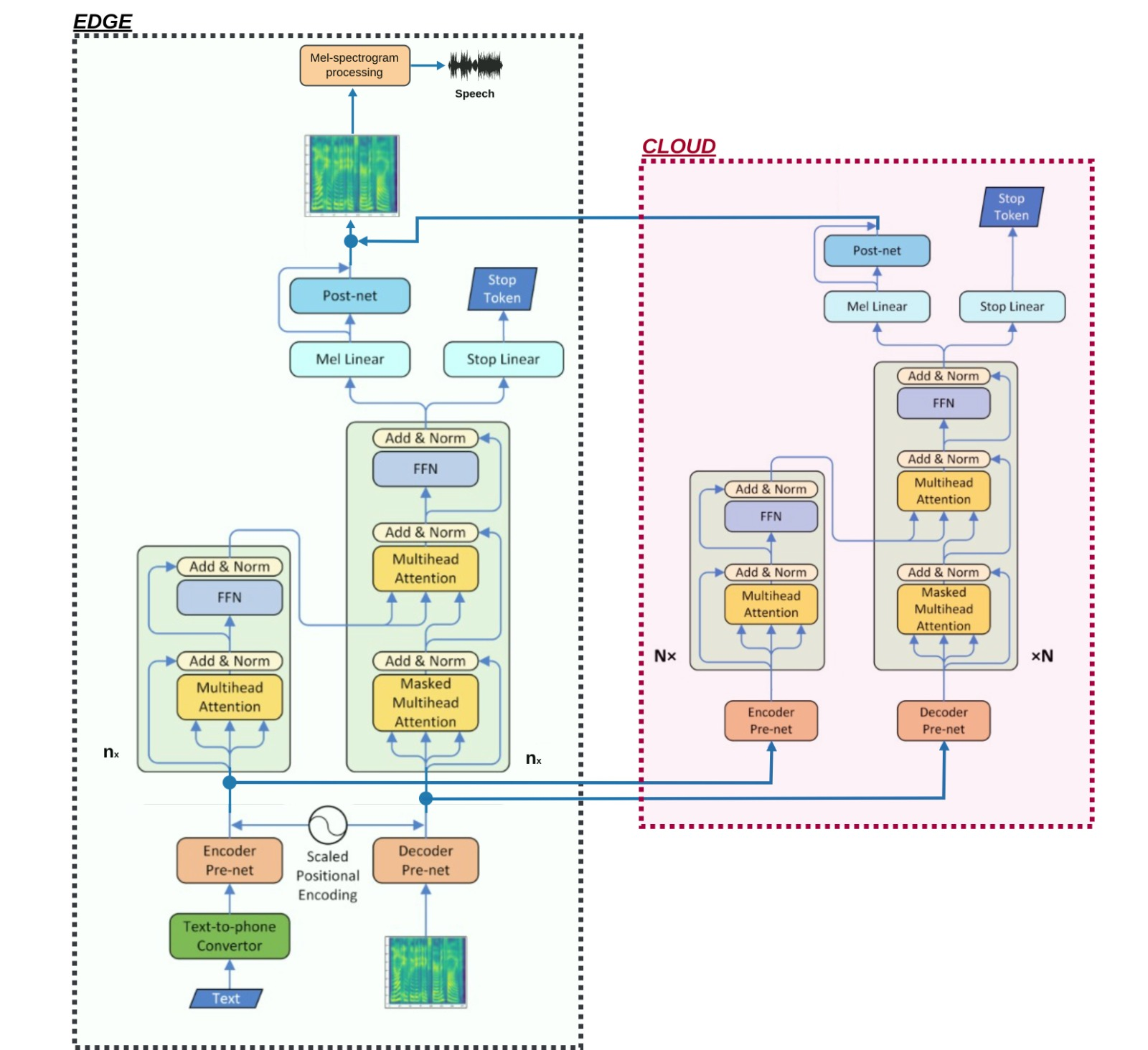
****

***Figure 1:Speech-To-Text Cascaded Architecture***

*In the figure above (Figure 1), a Cascaded STT model architecture is presented. As indicated in the figure, on the left, an edge-level architecture with nx layers is initially computed and produces output. When the output probabilities do not meet the set threshold, the initially computed representation of the input speech at the CNN level as well as its corresponding embedding representations will be sent to the cloud for further processing.*

1. **Text-To-Speech**

The Text-To-Speech model for the edge device adheres to an encoder-decoder structure. The encoder begins with an embedding module known as the “encoder pre-net,” which learns phoneme representations from input text. Through scaled positional encoding and N blocks of multi-head attention plus feedforward layers, the encoder produces phoneme representations to serve as a context for the decoder. The decoder, responsible for audio signal modeling (in the form of mel spectrograms), starts with a convolutional neural network layer (termed the “Decoder Pre-net''). It is followed by a masked multi-head attention mechanism, a cross-attention layer that incorporates features from the decoder, and a subsequent feed-forward network for mel-spectrogram prediction. The model further processes the output using a linear layer (“mel linear”) and a convolutional neural network (“post-net”). The predicted mel-spectrograms can be converted back into audio signals using the Griffin Lim algorithm (Griffin and Lim 1984). This edge-based text-to-speech model’s features would be sent to the cloud model when the average noise in the output audio exceeds a predefined threshold. This selective workload balancing benefits both the edge and the cloud by effectively distributing the entire inference process between the edge devices and the cloud’s computing resources.



***Figure 2:Speech-To-Text Cascaded architecture***

*In the figure above (Figure 2), a Cascaded TTS model architecture is presented. As indicated in the figure, on the left, an edge-level architecture with nx layers to be initially computed and produces output. When the output noise in the audio exceeds the threshold, the initially computed input text representation after the speech embedding as well as its corresponding speech representations will be sent to the cloud for further processing.*

**Data collection**

In the data collection phase of our research, we utilized the Mozilla Common Voice dataset[[1]](#footnote-0), a rich and diverse collection of voice recordings that is publicly available and widely recognized for its utility in speech technology research. We selected a subset dataset that comprised a total sample of 256 voice recordings, along with their corresponding textual transcriptions, for both Kiswahili and Kinyarwanda languages. To ensure the robustness and generalizability of our models, we divided the dataset into two distinct subsets: training and testing. The training set, which constitutes the majority of the data, is used to train our models, allowing them to learn the patterns and nuances of Kinyarwanda and Kiswahili speech. This process involves exposing the models to a wide array of vocal tones, dialects, and speech contexts, thereby enhancing their ability to accurately transcribe and synthesize speech in these languages. The testing set, on the other hand, serves a critical role in evaluating the performance of our models.

**Experiments and Results - Our own created model architecture**

1. **Speech To Text**

* **Model training**

The development of STT models for both Kinyarwanda and Swahili languages were developed and trained from scratch using Pytorch. The realization of this approach was conducted in three main steps. The first step was dedicated to data preprocessing which included the extraction of 27 Mel-Frequency Cepstral Coefficients (MFCCs) from audio data. This step is crucial as it constructs the feature representation of the entire speech dataset. After that, the ASR model was constructed. In this step, the constructed model was based on two distinct architectures, namely: a traditional LSTM and a modern transformer model.

The LSTM-based model employs Convolutional Neural Networks (CNNs) for initial feature extraction, followed by a series of LSTM and Pyramidal Bi-LSTM layers for temporal feature processing. On the other hand, the transformer-based model uses encoder and decoder layers for sequence-to-sequence learning mechanisms. Both LSTM and transformer-based models are equipped with a decoder component (an MLP in the case of the LSTM-based model) to map encoded features to the target vocabulary.

* **Training parameters:**

In our approach, the following parameters were used:

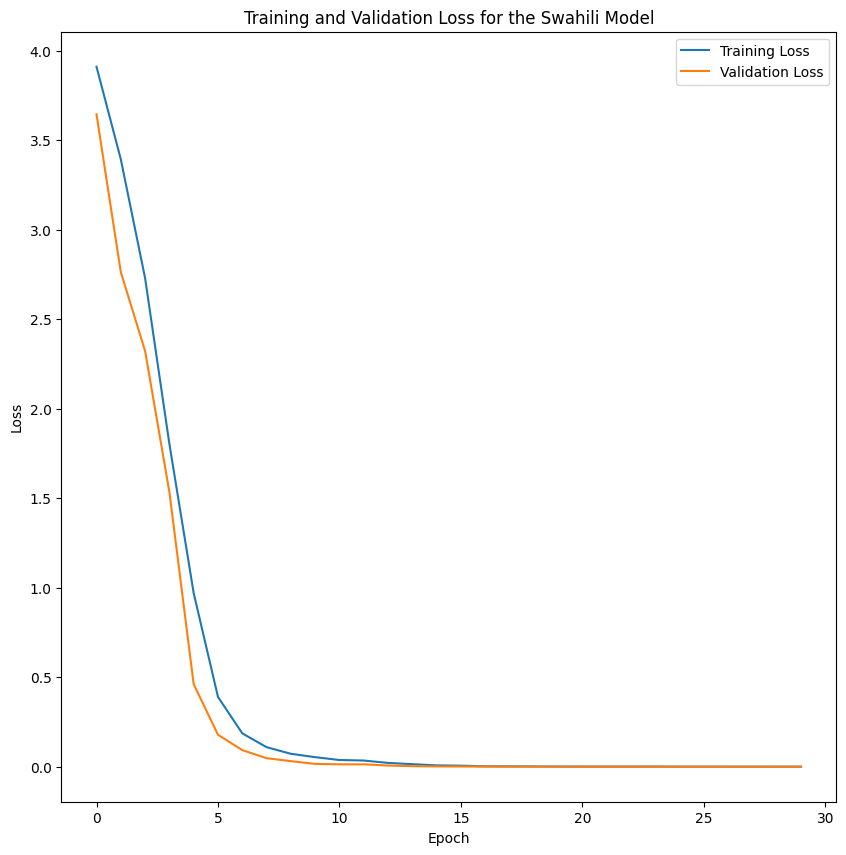
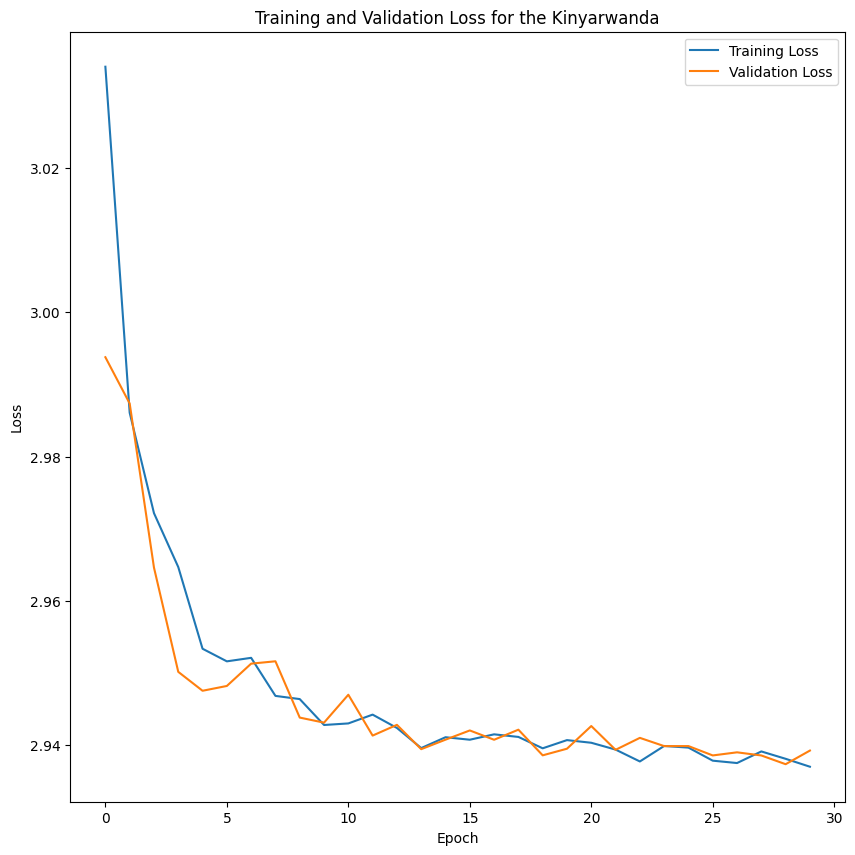
* The input features (with a dimension of 27 representing MFCC features) output a higher-dimensional space (embed\_size=256) for more effective representation.
* The embedding output is passed through a series of Transformer encoder and decoder layers with the following characteristics:
  + 4 attention heads (nhead=4), and
  + both the encoder and decoder consist of a single layer
  + The maximum sequence length is set to 500
  + The output size of 256 matches the length of the vocabulary.
* The hyperparameters were used during training:
  + Batch size:8
  + Learning late:2e-3
  + Number of epochs: 30
  + Optimizer: Adam
  + Loss: Cross Entropy loss
* **Evaluation Metric**

Character Error Rate (CER) was used during the evaluation process. It quantitatively measures the performance of an ASR model by comparing the transcription generated by the model against a reference or ground truth transcription. CER is calculated as the sum of substitutions, insertions, and deletions needed to transform the model output into the reference transcription, divided by the number of words in the reference. This metric provides a straightforward and objective means to gauge the accuracy of ASR systems. It also offers insights into how well a model understands and transcribes spoken language.

1. **Results**

The graphs presented illustrate the training and validation loss trends over 30 epochs for two distinct Automatic Speech Recognition (ASR) models, one for the Kinyarwanda language and the other for Kiswahili. For both models, the training loss and validation loss decrease sharply in the initial epochs, indicating rapid learning and model improvement. As the epochs progress, both losses plateau, suggesting that the models are approaching their optimal performance on the given data. The Kinyarwanda model exhibits a closer convergence between training and validation losses, which may imply better generalization when compared to the Kiswahili model. However, both models demonstrate a similar pattern of loss reduction, which is characteristic of effective learning dynamics in neural network training. The slight fluctuations in validation loss for the Kinyarwanda model towards the later epochs could be indicative of the model's sensitivity to the validation set's variability or minor overfitting.

The Kiswahili ASR model achieved a CER of 0.011, while the Kinyarwanda model recorded a slightly higher CER of 0.017. a lower CER is typically indicative of better performance, as it reflects fewer errors made by the ASR system in transcribing the spoken language. These results indicate both models were able to transcribe accurately. It is noteworthy that, within the scope of this research, no baseline models were found that utilize CER as a metric for these specific languages. This absence of a direct comparison highlights the novelty of this work and underscores the potential of these models to serve as benchmarks for future research in Kinyarwanda and Swahili ASR systems.



***Figure 3: Training and validation losses of both Kinyarwanda and Swahili ASR systems***

1. **Text To Speech**

* **Model training**

The model is composed of two primary components: an encoder and a decoder, both of which are built upon the self-attention mechanism that is central to Transformer models. The encoder is responsible for processing the input text sequence. It first embeds the input tokens using a learned embedding layer, which is then passed through a pre-net consisting of convolutional layers with batch normalization and dropout for regularization. The output of the pre-net is combined with positional embeddings to retain the sequence order information. This combined representation is then fed into a series of Transformer blocks, each comprising a multi-head self-attention module and a feed-forward network, with layer normalization and dropout applied to stabilize and regularize the training process.

The decoder takes the processed encoder output and the target mel-spectrogram as inputs. It employs a pre-net that transforms the mel-spectrogram before adding positional embeddings. The core of the decoder is a series of decoder blocks, each containing a causally masked self-attention layer to prevent future information leakage and a Transformer block to attend over the encoder outputs. The output of the decoder blocks is projected to the mel-spectrogram space and further refined by a post-net, which is a stack of convolutional layers designed to capture the local spectral features of the speech signal. This architecture is trained end-to-end, with the ability to generate a mel-spectrogram from text, which can then be converted into an audio waveform using a vocoder.

* **Training parameters:**

In our approach, the following parameters were used:

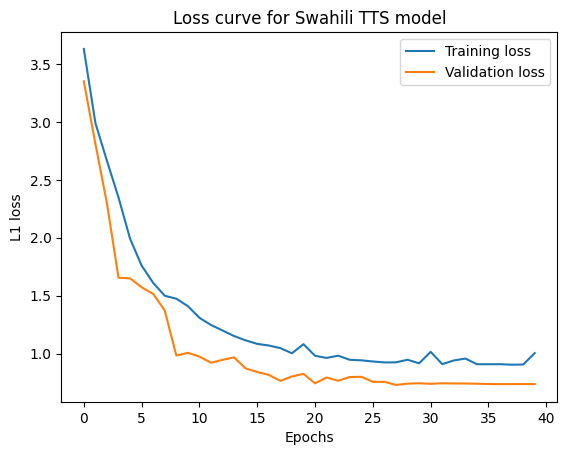
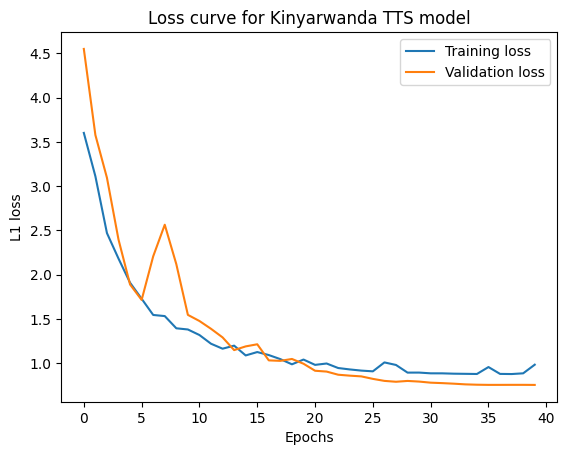
* The input features vocab\_size=30 outputting a higher-dimensional space (embed\_dims=512) for more effective representation.
* The output is passed through a series of encoder and decoder layers with the following characteristics:
  + hidden\_dims=256,
  + n\_heads=4,
  + forward\_expansion=4,
  + num\_layers=6,
  + mel\_dims=80,
  + dropout = 0.15
  + max\_len = 1024
  + pad\_idx = 0
* The hyperparameter were used during training:
  + Batch\_Size = 2
  + Epochs = 40
  + LR = 3e-4
  + warmup\_steps = 0.2
  + Optimizer : AdamW
  + Loss: BCEWithLogitsLoss,L1 Loss
* **Evaluation Metric**

Loss functions are fundamental in training and evaluating text-to-speech (TTS) models. They are crucial for optimizing model performance. Fu et al. (2018) emphasize the significance of metrics like mean square error (MSE) in optimizing TTS models [24]. Additionally, Martín-Doñas et al. proposed a perceptual metric for speech quality evaluation that can function as a loss function during deep learning model training [25]. These studies underscore the importance of using appropriate loss functions aligned with the desired outcomes of TTS models. By incorporating loss as an evaluation metric, researchers can effectively assess the performance of TTS models and refine them to improve speech quality and intelligibility. For our text-to-speech models, we used the mean square error (MSE) as the evaluation metric.

**Results**

The loss curves in the figure below provide insights into the training dynamics for the Kinyarwanda and Kiswahili Text-to-Speech (TTS) models over 40 epochs. As indicated, both models exhibit a sharp decline in training and validation loss during the initial epochs, which technically indicates rapid learning and adaptation to the training dataset. As the epochs progress, the rate of loss reduction slows, and the curves begin to plateau, which suggests model convergence towards their optimal performance.

Notably, the Kinyarwanda TTS model shows a slight divergence between training and validation loss after approximately 10 epochs, which implies a degree of overfitting to the training data. In contrast, the Kiswahili TTS model maintains a closer convergence between the training and validation losses throughout the training process, indicating better generalization to unseen data. These trends reflect the models' ability to learn and generate speech from text, with the observed differences highlighting the nuances of language-specific model training and optimization.



***Figure 4: Training and validation losses of both Kinyarwanda and Swahili TTS systems***

**Experiments and Results of Fine-tuned Models**

Since the results of our experiments did not meet the threshold of the state-of-the-art, the next step in our experiments was to fine-tune already existing and available pre-trained models for the Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) tasks and adapt them for the Kinyarwanda and Swahili languages.

1. **Speech-to-text**

In these experiments, we explored the development of a speech-to-text synthesis system for Kinyarwanda and Swahili languages. We leveraged the power of a pre-trained model called Whisper and fine-tuned it on Mozilla Common Voice datasets specific to the Kinyarwanda and Swahili languages.

Whisper, developed and released by OpenAI in 2022, is a transformer-based neural network architecture [26] designed for a variety of speech processing tasks, including automatic speech recognition (ASR) and speech translation. The main purpose of this model was to create an ASR model that ‘works reliably without the need for dataset-specific fine-tuning to achieve high-quality results on specific distributions’. It was trained on 680,000 hours of multilingual and multitasking-supervised data collected from the web. The Whisper architecture is based on an encoder-decoder transformer. This model enabled transcription in multiple languages as well as translation from those languages into English.

This pre-trained model boasts several key aspects:

* **End-to-End Design**: Whisper uses a simple and efficient end-to-end architecture. Audio input is converted into a log-Mel spectrogram, a visual representation of the audio's frequency content. This spectrogram is then fed into the encoder portion of the model. The decoder section is tasked with predicting the corresponding text transcript, along with additional information like spoken language identification and timestamps. This unified approach eliminates the need for complex pipelines often used in traditional speech processing systems.
* **Multilingual and Multitask Learning:** Whisper is trained on a massive dataset of diverse audio recordings in multiple languages. This allows the model to handle various speech characteristics, like accents and background noise, while simultaneously performing multiple tasks like speech recognition and language identification.
* Open-Source Availability: OpenAI has made Whisper publicly available, allowing researchers like ourselves to leverage its capabilities and fine-tune it for specific applications.

These characteristics made Whisper an ideal pre-trained model for our text-to-speech synthesis project in Kinyarwanda and Swahili. By fine-tuning Whisper on language-specific datasets, we can specialize its capabilities for high-quality speech generation in these target languages.

The Kinyarwanda and Swahili datasets from Mozilla Common Voice provided us with the essential speech samples required for fine-tuning the Whisper model. These datasets encompass a diverse range of speakers, ensuring the model is exposed to various speech patterns and pronunciations specific to each language.

**Evaluation Metric: Word Error Rate (WER)**

To assess the performance of our fine-tuned model, we used the Word Error Rate (WER) metric. WER calculates the number of errors (insertions, substitutions, and deletions) an automatic speech recognition system makes when compared to a human-generated reference transcript. In the context of speech-to-text transcription, a lower WER indicates a higher fidelity between the synthesized speech and the original written text.

**Results**

The Whisper model was fine-tuned for each language independently. After 11 hours of training on the Swahili Common Voice dataset, our model achieved a WER of 34.9. This was a significant improvement over OpenAI's previously reported WER of 51.2 for Swahili speech-to-text transcription [27]. A live demonstration of the Swahili automatic speech recognition fine-tuned model can be found on the hugging face platform at:

<https://huggingface.co/NMutangana/whisper-small-swahili> [28].

For Kinyarwanda, the fine-tuned model required 15 hours of training on the Common Voice dataset. The resulting WER was 68.7. While this is higher than the Swahili result, it represents a promising initial step for Kinyarwanda speech-to-text development using this methodology. A live demonstration of the Kinyarwanda automatic speech recognition fine-tuned model can be found on the hugging face platform at: <https://huggingface.co/NMutangana/whisper-small-rw> [29].

These results, detailed in Table 1 below, showcase the potential of fine-tuning pre-trained models like Whisper on language-specific datasets to create functional speech-to-text systems for under-resourced languages like Kinyarwanda and Swahili.

| **Swahili Fine-tuned Whisper Model Results** | | | | **Kinyarwanda Fine-tuned Whisper Model Results** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Training Loss** | **Validation Loss** | **Wer** | **Step** | **Training Loss** | **Validation Loss** | **Wer** |
| 1000 | 0.379700 | 0.602877 | 40.236108 | 1000 | 0.695200 | 0.991951 | 70.502285 |
| 2000 | 0.112900 | 0.588555 | 35.584063 | 2000 | 0.477100 | 0.926026 | 68.362372 |
| 3000 | 0.050700 | 0.639749 | 35.440375 | 3000 | 0.313200 | 0.950552 | 68.073194 |
| 4000 | 0.016100 | 0.682069 | **34.943789** | 4000 | 0.202500 | 0.991832 | **68.727057** |

***Table 1: Results after Finetuning the Whisper Automatic Speech Recognition Model on Swahili and Kinyarwanda Datasets.***

1. **Text-to-Speech**

In these experiments, we explored an alternative approach to text-to-speech synthesis for Kinyarwanda and Swahili languages. Here, we leveraged the capabilities of a pre-trained model called SpeechT5, fine-tuned on the Mozilla Common Voice datasets for these languages.

SpeechT5, introduced by Ao et al. (2021), is a unified pre-trained model designed for various spoken language processing tasks [30]. This model builds upon the success of T5 (Text-to-Text Transfer Transformer) but extends its capabilities to the speech domain. Below is a detailed description of SpeechT5's key features:

* **Unified Encoder-Decoder Architecture:** SpeechT5 employs a shared encoder-decoder architecture to handle both speech and text modalities. This allows the model to learn a unified representation of spoken and written language, facilitating tasks like text-to-speech synthesis.
* **Pre-Trained on Massive Datasets:** Similar to Whisper, SpeechT5 is pre-trained on a large corpus of speech and text data. This equips the model with a strong foundation for understanding linguistic structures and generating natural-sounding speech in various languages.
* **Fine-Tuning Potential:** SpeechT5's pre-trained capabilities serve as a valuable starting point for further specialization. By fine-tuning on language-specific datasets like Kinyarwanda and Swahili Common Voice, we can enhance the model's ability to produce high-quality speech in these target languages.

**Datasets**

We fine-tuned SpeechT5 on the Kinyarwanda and Swahili Common Voice datasets independently. It's important to acknowledge that Common Voice datasets, while valuable for ASR training due to the presence of real-world noise, aren't ideal for TTS purposes. This is because speech characteristics desirable for ASR (e.g., diverse accents and background noise) can hinder the naturalness of synthesized speech. In an ideal scenario, high-quality, multilingual, multi-speaker TTS datasets specifically designed for clean speech would be preferred. However, due to the scarcity of such resources for under-resourced languages like Kinyarwanda and Swahili, Common Voice datasets have become a practical alternative. We acknowledge this limitation and aim to explore superior datasets in future research.

**Evaluation Metric: Validation Loss**

In this text-to-speech exploration, we employed validation loss as a performance indicator. During model training, the model is exposed to both training and validation data. The training data is used to adjust the model's internal parameters, while the validation data helps assess how well the model generalizes to unseen examples. Validation loss measures the model's performance on the validation data. Lower validation loss signifies better model performance during fine-tuning, indicating the model is learning to generate speech that aligns well with the ground truth (written text).

**Results**

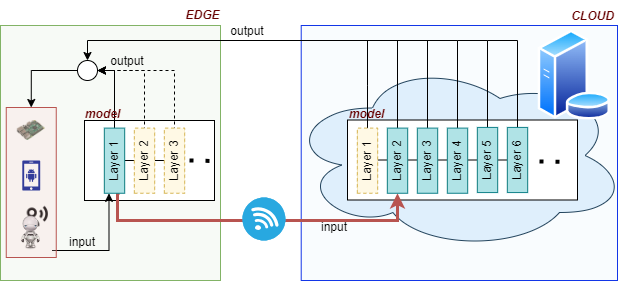
After 21 epochs of training, the validation loss for the Swahili TTS model was 0.523976, and for the Kinyarwanda TTS model, it was 0.497876. While these values don't directly translate to speech quality, lower validation loss generally suggests the model is on track to generate more natural-sounding speech. The training results are detailed below in Table 2.

| **Swahili Fine-tuned SpeechT5 Model Results** | | | **Kinyarwanda Fine-tuned SpeechT5 Model Results** | | |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Training Loss** | **Validation Loss** | **Step** | **Training Loss** | **Validation Loss** |
| 1000 | 0.598900 | 0.553199 | 1000 | 0.695200 | 0.991951 |
| 2000 | 0.564900 | 0.534779 | 2000 | 0.477100 | 0.926026 |
| 3000 | 0.562600 | 0.526816 | 3000 | 0.313200 | 0.950552 |
| 4000 | 0.556600 | **0.523976** | 4000 | 0.533400 | **0.497876** |

***Table 1: Results after Finetuning the SpeechT5 Text-to-Speech Model on Swahili and Kinyarwanda Datasets***

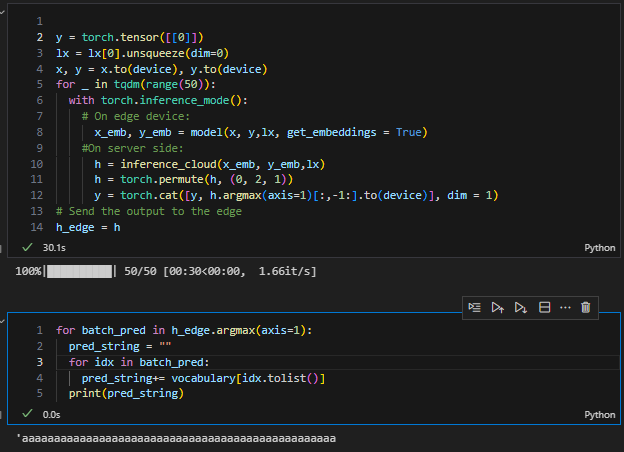
**Deployment on Edge**

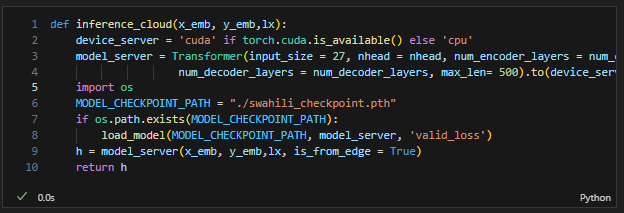
In our research, we introduce an innovative architecture for deploying text-to-speech and speech-to-text models called “the cascading architecture”, which leverages both edge and cloud resources. The architecture comprises two distinct models: a lightweight model deployed at the edge and a more sophisticated model hosted in the cloud. The figure below illustrates a generic cascading model, where from any layer of the edge model, engineered features can be sent to the cloud model, and enhanced predictions can be drawn back from the cloud to the edge.

****

**Figure:**The network begins processing on the constrained device and only relies on the remote part when the local part does not provide an accurate enough result.

For our scenario, as it has been shown in Figure 1 and Figure 2, the edge model is responsible for initial predictions and can generate embeddings from raw inputs (text for text-to-speech and speech for speech-to-text). However, when the edge model produces suboptimal predictions, it forwards the embeddings to the cloud model. The cloud model, being more powerful, refines these embeddings and generates higher-quality outputs. Finally, the refined outputs are transmitted back to the edge device. This approach ensures efficient real-time processing while maintaining accuracy, making it suitable for resource-constrained edge devices.

****

****

**On Cloud (Server-side)**

**On Edge (Client-side)**

**Future work and Conclusion**

The paper presents a novel approach for edge-based speech transcription and synthesis targeting the Kinyarwanda and Swahili languages leveraging cascading network architecture. This system processes speech data locally on edge devices and escalates to cloud-based resources when necessary, facilitating efficient use of computational resources and enhanced accessibility. Key advantages of this approach include the efficient handling of local language processing on low-resource devices, facilitating real-time speech processing, and reducing reliance on cloud compute services.

Although the benefits of cascading techniques are enormous, several challenges and limitations arise. One significant challenge is the high bandwidth requirement for edge-to-cloud connection, which is essential for real-time data processing and decision-making. Another issue lies in determining the event that triggers the need to escalate tasks to the cloud, a process that often requires experimentation to optimize. Additionally, the training phase is constrained by a small dataset, typically due to limited available resources. For future work, it is crucial to focus on determining the escalation trigger accurately and efficiently, as well as exploring suitable data compression techniques for more efficient communication. Furthermore, expanding the training dataset to a larger scale can lead to improved model performance and generalization.

**References**

[1]“Swahili | Critical Languages Program | College of Humanities.”

https://clp.arizona.edu/courses/languages/swahili

[2]“Rwanda’s population reaches in 2022 | National Institute of Statistics Rwanda.” https://www.statistics.gov.rw/publication/Rwanda\_population\_2022

[3] “AwezaMed: A Multilingual, Multimodal Speech-To-Speech Translation Application for Maternal Health Care,” IEEE Conference Publication | IEEE Xplore, Jul. 01, 2020. https://ieeexplore.ieee.org/document/9190240

[4] D. C. T. Metalom, J. L. F. K. Ebongue, and B. O. Yenke, “Towards to a Direct Speech to Speech for Endangered Languages in Africa,” Oct. 04, 2022. https://inria.hal.science/hal 03711256/

[5] L. Silver, “1. Majorities in sub-Saharan Africa own mobile phones, but smartphone adoption is modest,” Pew Research Center’s Global Attitudes Project,

https://www.pewresearch.org/global/2018/10/09/majorities-in-sub-saharan-africa-own-mobile phones-but-smartphone-adoption-is-modest/.

[6] K. Wei et al., "Joint Pre-Training with Speech and Bilingual Text for Direct Speech to Speech Translation," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi:

10.1109/ICASSP49357.2023.10095616.

[7] X. Li, Y. Jia and C. -C. Chiu, "Textless Direct Speech-to-Speech Translation with Discrete Speech Representation," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096797.

[8] X. -P. Nguyen, S. Popuri, C. Wang, Y. Tang, I. Kulikov and H. Gong, "Improving Speech-to Speech Translation Through Unlabeled Text," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095578.

[9] Jia, Y., Ramanovich, M. T., Wang, Q., & Zen, H. (2022). CVSS Corpus and Massively Multilingual Speech-to-Speech Translation. ArXiv. /abs/2201.03713

[10] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter and S. Watanabe, "End-to-End Speech Recognition: A Survey," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, pp. 325-351, 2024, doi: 10.1109/TASLP.2023.3328283.

[11] A. Gulati et al., “Conformer: Convolution-augmented transformer for speech recognition,” Interspeech 2020, Oct. 2020. doi:10.21437/interspeech.2020-3015

[12] P. Guo et al., "Recent Developments on Espnet Toolkit Boosted By Conformer," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 5874-5878, doi:

10.1109/ICASSP39728.2021.9414858.

[13] W.-N. Hsu et al., “Hubert: Self-supervised speech representation learning by masked prediction of Hidden Units,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 3451–3460, 2021. doi:10.1109/taslp.2021.3122291

[14] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, “End-to-End ASR: from Supervised to Semi-Supervised Learning with Modern Architectures,” in Proc. ICML, Jul. 2020.

[15] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T. (2020). FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. ArXiv. /abs/2006.04558

[16] R. Atienza, "EfficientSpeech: An On-Device Text to Speech Model," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10094639.

[17] Y. Wang et al, "Tacotron: Towards End-to-End Speech Synthesis," ArXiv.Org, 2017. Available: https://www.proquest.com/working-papers/tacotron-towards-end-speech synthesis/docview/2075135809/se-2.

[18] J. Shen et al., "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 4779-4783, doi: 10.1109/ICASSP.2018.8461368.

[19] R. Prenger, R. Valle and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 3617-3621, doi: 10.1109/ICASSP.2019.8683143.

[20] Y. He et al., "Streaming End-to-end Speech Recognition for Mobile Devices," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 6381-6385, doi: 10.1109/ICASSP.2019.8682336.

[21] Asiedu Asante, Bismark, and Hiroki Imamura. “Speech Recognition and Speech Synthesis Models for Micro Devices.” ITM Web of Conferences, vol. 27, 2019, pp. 5001-, https://doi.org/10.1051/itmconf/20192705001.

[22] J. Chen, X. Song, Z. Peng, B. Zhang, F. Pan and Z. Wu, "LightGrad: Lightweight Diffusion Probabilistic Model for Text-to-Speech," ICASSP 2023 - 2023 IEEE International Conference on

Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096710.

[23] Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. LRSpeech: Extremely Low-Resource Speech Synthesis and Recognition. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining (KDD '20). Association for Computing Machinery, New York, NY, USA, 2802–2812. <https://doi.org/10.1145/3394486.3403331>

[24] Fu, S., Wang, T., Tsao, Y., Lu, X., & Kawai, H. (2018). End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks. Ieee/Acm Transactions on Audio Speech and Language Processing, 26(9), 1570-1584. https://doi.org/10.1109/taslp.2018.2821903

[25] Martín-Doñas, J., Gomez, A., González, J., & Peinado, A. (2018). A deep learning loss function based on the perceptual evaluation of the speech quality. Ieee Signal Processing Letters, 25(11), 1680-1684. <https://doi.org/10.1109/lsp.2018.2871419>

[26] A. Radford, J. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision.” Available: <https://cdn.openai.com/papers/whisper.pdf>

[27] OpenAI, “Whisper,” GitHub, Oct. 09, 2022. <https://github.com/openai/whisper>

[28] “NMutangana/whisper-small-swahili · Hugging Face,” huggingface.co. <https://huggingface.co/NMutangana/whisper-small-swahili>.

‌[29] “NMutangana/whisper-small-rw · Hugging Face,” huggingface.co. <https://huggingface.co/NMutangana/whisper-small-rw>.

[30] J. Ao et al., “SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing.” Available: <https://arxiv.org/pdf/2110.07205.pdf>

[31] **Github Source code**:

<https://github.com/pacomesimon/Edge-Based-Speech-Transcription-and-Synthesis/tree/main>

1. <https://huggingface.co/datasets/mozilla-foundation/common_voice_13_0> [↑](#footnote-ref-0)