Journal of Applied Artificial Intelligence
https://jaai.sabapub.com
ISSN: 2709-5908

2025 Volume 6, Issue 1: 39-52
DOI: https://doi.org/10.48185/jaai.v6i1.1474

YorubaAI: Bridging Language Barrier with Advanced Language Models

Kamoli Akinwale Amusa[1,*], Tolulope Christiana Erinosho[2], Olubusola Olufunke Nuga[3], Abdulmatin Olalekan Omotoso[4]

[1,2,3,4]Electrical and Electronics Engineering Department, College of Engineering, Federal University of Agriculture, Abeokuta 110213, Nigeria

Received: 24.01.2024 • Accepted: 02.04.2025 • Published: 16.05.2025 • Final Version: 30.05.2025

Abstract: YorubaAI addresses the digital divide caused by language barriers, particularly for Yoruba language speakers who struggle to interact with advanced large language models (LLMs) like GPT-4, which primarily support high-resource languages. This study develops a system, named YorubaAI, for seamless communication in Yoruba language with LLMs. The YorubaAI enables users to input and receive responses in Yoruba language, both in text and audio formats. To achieve this, a speech-to-text (STT) model is fine-tuned for automatic Yoruba language speech recognition while a text-to-speech (TTS) model is employed for conversion of Yoruba language text to speech equivalent. Direct communication with LLM in low-resource languages like Yoruba language typically yields poor results. To prevent this, a generation technique known as retrieval-augmented generation (RAG) is utilized to augment the LLM's existing knowledge with additional information. The RAG is formed through creation of a database of questions and answers in Yoruba language. This database serves as the primary knowledge base that the YorubaAI uses to retrieve relevant information with respect to the question asked. The content of the created questions and answers database is converted into vector embeddings using Google's Language-Agnostic BERT Sentence Embedding (LaBSE) model to yield numerical representations that capture the semantic meaning of the texts. The embeddings generated from the Yoruba questions database are stored in a vector store database. These embeddings were essential for efficient search and retrieval.The the two models (STT and TTS models) were integrated with a LLM using a user-friendly interface that was built using the Gradio framework. The STT model achieved a word error rate of 13.06% while the TTS model generated natural-sounding Yoruba language speech. YorubaAI correctly responded to various queries in pure Yoruba language syntax and thus successfully bridges the AI accessibility gap for Yoruba language speakers.

Keywords: Yoruba language, speech-to-text, text-to-speech, natural language processing, low-resource language

1. Introduction

Natural language processing (NLP) is a field of study that uses artificial intelligence (AI) and machine learning to teach computers to understand and communicate with human language. NLP is a key technology in AI that allows computers to: Interpret human language, analyze human language, and generate human language. With the vast knowledge and opportunities unlocked by advanced large language models (LLMs), a type of artificial intelligence (AI) system designed to understand, generate, and work with human language in a way that mimics how humans communicate, it is still inaccessible

* Corresponding Author: amusaka@funaab.edu.ng

to millions of people worldwide because these wonders speak a language foreign to them. In a world where it is estimated that over 7000 languages are spoken, the digital divide widens not just by access to technology but by the languages that the technology understands and speaks. The introduction of LLMs promised a future of limitless knowledge sharing and problem-solving, yet for speakers of languages such as Yoruba, this future remains just beyond the reach.

Language plays a fundamental role in facilitating communication and self-expression for humans, and their interaction with machines [1]. Communication, in its essence, is about connection and information sharing. It is the transmission of information between individuals, and it can take various forms such as verbal, non-verbal, and written exchanges. It plays a crucial role in the society, enabling the sharing of thoughts, opinions, and knowledge. Communication can occur through different channels, including mass media, interpersonal interactions, and technological advancements like the Internet.

Language is a prominent ability in human beings to express and communicate, which develops in early childhood and evolves over a lifetime. Machines, however, cannot naturally grasp the ability to understand and communicate in the form of human language, unless equipped with powerful AI algorithms. It has been a longstanding research challenge to achieve this goal, to enable machines to read, write, and communicate like humans. Technically, language modeling (LM) is one of the major approaches to advancing the language intelligence of machines [2].

Different studies have been carried out concerning the development of LMs, LLMs, and Automatic Speech Recognition (ASR) systems both for high-resource and low-resource languages, and language translation, which are all germane to the YorubaAI. Few of such efforts include transformer model which utilizes a self-attention mechanism to weigh the significance of different words in a sentence [3], semi-supervised generative pre-training of a language model and its subsequent task-specific fine-tuning [4], Bidirectional Encoder Representations from Transformers (BERT) for NLP tasks [5], transformer-transducer for ASR [6], FastSpeech series [7, 8], transformer-based acoustic models for hybrid ASR systems [9-11], conformer for ASR tasks [12], AfriBERT, a BERT adaptation for Afrikaans [13], Neural Machine Translation (NMT) for low-resourced languages [14-15], optimized transformers for low-resource NMT tasks [16].

Other notable developments are combined Recursive Neural Networks (RNNs) and transformers for translation of similar language pairs [17], character-level NMT [18], mT5: multilingual pre-trained Text-To-Text Transfer Transformer [19], a diffusion-based model for TTS aligning text with audio using stochastic differential equations [20], Whisper, an ASR model trained on 680,000 hours of multilingual audio that achieves near-human performance without fine-tuning [21], a model for translating English number texts into Yoruba [16], Generative Pre-trained Transformer-4 (GPT-4), a multimodal large language model capable of processing text and images [22], multilingual speech recognition systems for African languages [13], the universal speech model [23], Zipformer, a more efficient alternative to Conformer [24] and a morphology-aware BERT for TTS [25-26].

In spite of the significant advancement in LLMs [6, 7] such as Google's BERT [3], T5 [4], Robustly optimized BERT approach (RoBERTa) [27], and the GPT series such as GPT-3 and GPT-4 [22], much is still needed to be done on these LLMs in respect of communicating the same way humans do and perform various NLP tasks including translation from one language to another. These models only work well with high-resource languages like English, French, and German which makes people who speak low-resource languages like Yoruba not to be able to leverage the powerful capability of LLMs and use them for their various tasks. The promise of inclusivity remains unfulfilled for low-resource

language speakers. This barrier not only limits access to information but also stifles cultural expression in the digital realm.

Combining various TTS and STT models with these LLMs can help reduce the impacts of the language barrier on low-resource language speakers in accessing LLMs which in turn will allow them to benefit from these amazing advancements of the AI. This study, YorubaAI, is motivated by the need to liberalize access to AI technologies by ensuring language does not serve as an impediment but as a bridge to innovation, knowledge, and global participation, with a focus on Yoruba language which is one of the most widely spoken languages in Nigeria.

Despite Nigeria's large Yoruba-speaking population, there is currently no AI system allowing audio-based interactions with LLMs in Yoruba language, at least to the best of authors' knowledge. This creates a gap that hinders access to educational and technological opportunities for millions of Yoruba speakers. YorubaAI seeks to address this gap through creation of a system that allows Yoruba language speakers to use LLMs, such as ChatGPT and BART, through Yoruba language audio input and output, thus ensuring equitable access to these technologies. To that end, the development involves fine-tuning of a STT model for accurate Yoruba speech transcription, fine-tuning of a TTS model to generate natural Yoruba speech from the transcribed Yoruba speech and lastly creation of a user friendly interface to integrate both fined-tuned STT and TTS models with a LLM for real-time Yoruba language interaction.

2. Method

Approach employed in this study is sectionalized into six. Each of them is discussed in what follows beginning with overview of the proposed Yoruba AI system.

2.1. System overview

Figure 1 illustrates the block diagram of the YorubaAI system. The AI voice assistant system operates through a sophisticated pipeline which is designed to accept audio inputs in Yoruba language and produce audio output in the same format. The system's architecture comprises of few key components that are designed to perform specific tasks.

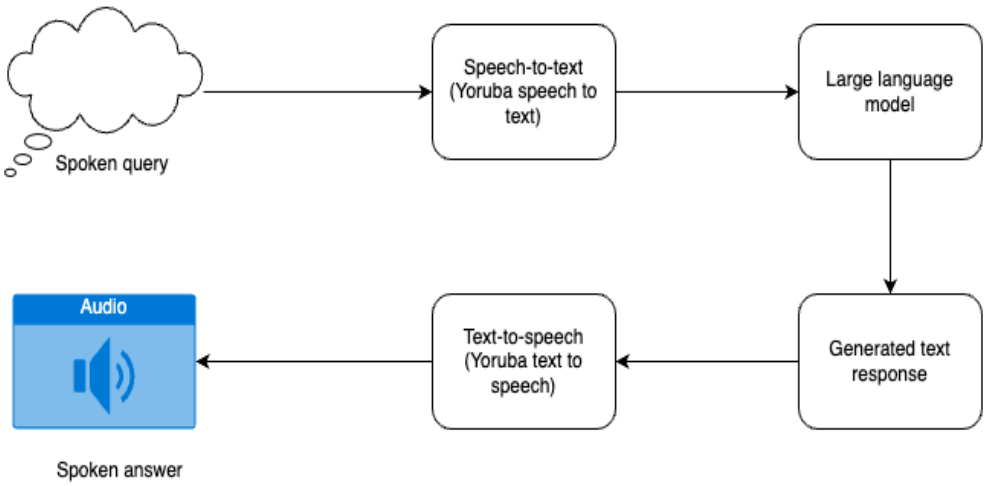


Figure 1. Block diagram of the YorubaAI system

The process begins when a user issues a spoken query, which is captured by the system with the help of the microphone, the spoken Yoruba query is transcribed into text via a speech-to-text component. This transcribed text is then passed to the large language model allowing the large language model to generate an appropriate response based on the query. Finally, the generated Yoruba text is converted into spoken audio through a text-to-speech system, delivering the answer audibly back to the user. This flow ensures the entire interaction occurs in Yoruba language.

2.2 Materials and tools

Hardware resources employed include high-performance computers and servers for processing and deploying of models as well as running of the application. Microphone is used for capturing of high-quality audio inputs. This means that the device or personal computer where this application will be running on must have a functioning microphone in place.

Python, a high-level programming language which has extensive libraries and frameworks is used for models building and data processing while Google Colab is employed as the environment for model training. For implementation, the Hugging Face Transformers libraries are utilized [28]. The choice is informed due to availability of several models which are trained on vast amount of data and on expensive computational resources on Hugging Face. These models (known as pre-trained models) can be easily fine-tuned on small datasets for a particular task and still achieve a very good accuracy. PyTorch is another open-source machine learning framework which is mostly used for natural language processing that is employed in this study. Gradio, an open-source Python framework for machine learning and data science teams [29], to quickly build demos or web application for machine learning models, is utilized in the creation of an interactive interface for YorubaAI system. This interface binds all the transformer's models used together to allow seamless interaction with the system by users.

2.3 Speech-to-text model for automatic Yoruba language speech recognition

The speech recognition component of the YorubaAI system was trained using the Common Voice 13 dataset [30]. This extensive dataset includes MP3 audio files paired with corresponding text files, encompassing a wide range of demographic metadata such as age, sex, and accent. This metadata is crucial as it allows the model to adapt to diverse speech patterns, thereby enhancing the accuracy of recognizing spoken Yoruba language speech. The dataset includes 27,141 recorded hours across 108 languages, with 17,689 validated hours, making it one of the most comprehensive resources for training speech recognition models.

The dataset is loaded into the notebook using the datasets API provided by Hugging Face for pre-processing. Since Yoruba is classified as a very low-resource language, the training and validation splits were combined to give approximately 6 hours of training data while the remaining 3 hours of test data was used as the held-out test set. Additional metadata information contained in Common Voice 13 dataset, such as accent and locale were disregarded for ASR. In other word, only the audio input and transcribed text were used for ASR fine-tuning. Other activities done in data pre-processing was down-sampling of audio inputs. Each of the audio input in the dataset is down-sampled to 16 kHz from 48 kHz that was used its preparation. This was done to ensure compatibility with sampling rate of 16 kHz required by the Whisper model that was used for feature extraction and ASR. After down-sampling, audio samples longer than 30 s were filtered out of the dataset. The filtering was done to avoid truncation that might result from such audio inputs when fed into the Whisper feature extractor, which could affect the stability of training.

After the pre-processing step, feature extraction followed which is done using the Whisper feature extractor. The feature extractor transforms the normalized audio signals into a log-Mel spectrogram. These spectrograms provide a time-frequency representation of the sound, with frequencies expressed on a Mel scale and amplitudes in decibels. Fine-tuning of the Whisper model for speech recognition on the Common Voice 13 dataset followed. A small version of the Whisper model and Yoruba split of the Common Voice 13 dataset are used. This allowed the fine-tuning to run quickly on any 16GB+ GPU with low disk space requirements such as the 16GB T4 GPU provided in the Google Colab free tier. The Google Colab notebook was linked to the Hub by simply entering the authentication token in order to ensure the trained model is uploaded directly to the hub.

A data collator was defined for the sequence-to-sequence speech model which was unique in the sense that the input features and labels were independently treated. While the input features were handled by the feature extractor, the labels were managed by the tokenizer. The input features were already padded to 30 s and converted to a log-Mel spectrogram of fixed dimension. The converted input features were then converted into batched PyTorch tensors. This was done using the feature extractor's .pad method with return_tensors=pt.

Since labels were un-padded. Label sequences were padded to the maximum length in the batch using the tokenizer's .pad method. The padding tokens were then replaced by (-100) so that these tokens were not taken into account when computing the loss. The start of the transcript token was then cut from the beginning of the label sequence and it was appended later during training. Next, the evaluation metric for fine-tuned STT model was defined. For this study, the Word Error Rate (WER) was chosen. This stage was followed by loading of pre-trained Whisper small checkpoint. The WER is expressed as:

$$WER = \frac{(S+I+D)}{N} \tag{1}$$

where (S, I, D) are substitutions, insertions and deletions, respectively. The value of WER cannot be a negative number but it could be above 100%.

In the final step, all parameters related to training were defined. Such parameters include the number of training steps, which was set to 500. This was enough steps to observe a significant WER improvement compared to the pre-trained Whisper model, while ensuring that fine-tuning could be run very quickly on the Google Colab free tier. The training was initialized by calling the train method, and the model, along with the results, was pushed into the hub.

2.4 Retrieval augmented generation database

Direct communication with LLM in low-resource languages like Yoruba typically yields poor results. To prevent this, we employ a generation technique known as retrieval-augmented generation. This technique will augment the LLM's existing knowledge with additional information. Figure 2 illustrated the retrieval augmented generation (RAG) method used for YorubaAI system.

The system began with a database of Yoruba questions and corresponding answers. This database served as the primary knowledge base that the YorubaAI system used to retrieve relevant information with respect to the question asked. The content of the Yoruba questions database was converted into vector embeddings using Google's Language-Agnostic BERT Sentence Embedding (LaBSE) model [31], which is trained on a diverse set of languages, including the Yoruba language. This process transformed the text data into a numerical representation that captured the semantic meaning of the text. These embeddings were essential for efficient search and retrieval.

The embeddings generated from the Yoruba questions database were stored in a vector store database. This specialized database was optimized for storing and querying high-dimensional vectors, making it efficient to retrieve similar or relevant embeddings based on a query. When a user provided a spoken query in Yoruba language, the system first transcribed the speech into text using a speech-to-text model. This transcribed query became the input for the retrieval process.

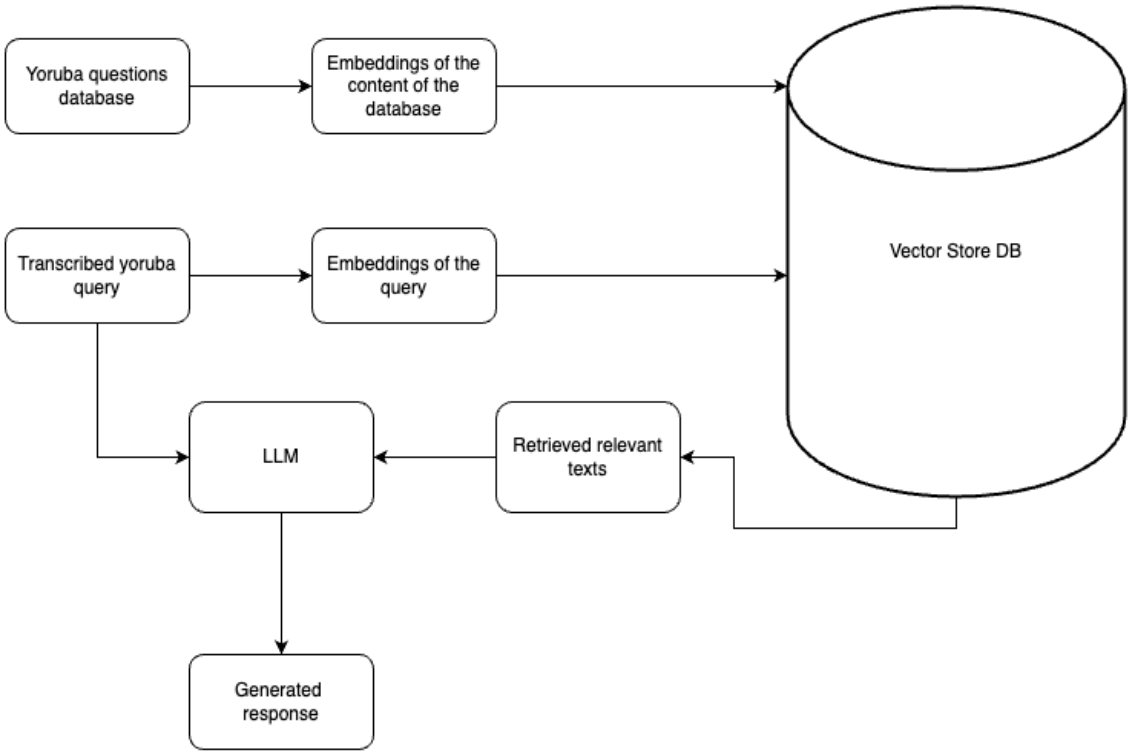


Figure 2. Block diagram of the RAG system

Similar to the database content, the transcribed Yoruba query would also be converted into an embedding. This embedding represented the query in the same vector space as the database content, allowing for similarity comparison. The embedding of the transcribed query would then be used to search the vector store database. The system retrieved the most relevant texts from the database based on the similarity between the query embedding and the database embedding.

The vector store database returned the most relevant texts that matched the query. These texts contained the information that would be used in the generation of a more accurate and contextually relevant response. The retrieved texts were passed to the LLM. The LLM used these retrieved texts, along with its internal knowledge, to generate a coherent and contextually accurate response to the user's query.

2.5 Text-to-speech model for conversion of English text to Yoruba speech

The TTS synthesis for Yoruba was developed using a transformer-based model fine-tuned on audio and corresponding text data in Yoruba. This model converts textual representations back into audio form, mimicking natural human speech. The model was not fine-tuned from the scratch, an existing model provided by Facebook which had been trained on Yoruba TTS was adopted. The model is open-source and is readily available on the Hugging face hub.

2.6 Interface for binding of fine-tuned models

To provide a seamless and user-friendly interface for interacting with the YorubaAI system, the Gradio library was employed, a powerful tool for building and deploying interactive web applications directly from Python scripts. Gradio was chosen for its simplicity, rapid development capabilities, and its ability to handle data-intensive applications efficiently. The interface design was centered on simplicity and functionality. The main page of the application features a minimalist layout which consist of two sections, one for the input and the other for the output. The input section was designed to take input in two forms: live audio recording using the microphone and uploading of a pre-recorded audio file. The output section was designed to display the transcribed question asked in Yoruba, the generated response to the question asked and the audio output of the generated response.

The Gradio interface acts as the frontend that binds all the backend functions into a unified system. Each of the components of the system (STT, RAG and TTS) are wrapped into a python function and the output of each function were passed to the other in a sequential manner. In other word, the output of the STT function fed the RAG function and finally the TTS function. The whole process was segmented into three sub-steps, briefly described as:

i. Speech transcription - Once activated, the subsequent audio is captured and sent to the speech transcription model, which converts the spoken Yoruba into text.

ii. Response generation-The transcribed text is then passed to the LLM to generate the corresponding answer for the query.

iii. Speech synthesis - The final Yoruba response is converted to speech and played back through the interface, allowing the user to hear the response.

The interface method in Gradio was then defined to give the system the required functionalities. Parameters such as the input formats needed (microphone and upload), outputs formats (textboxes and audio output), and the functions that bound all the components together were defined and passed to the interface method. The Gradio interface was launched and tested on Google Colab before it was finally deployed on HuggingFace hub.

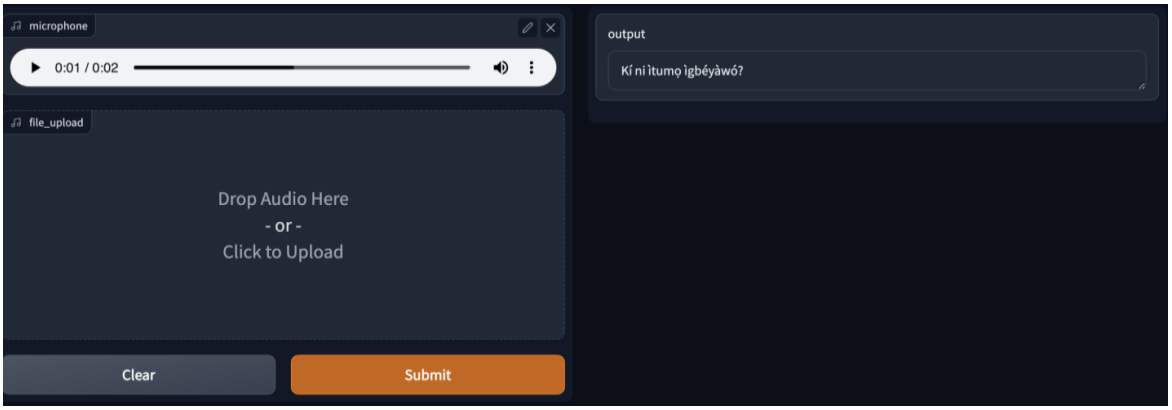
3. Results and discussion

The results section is split into three namely: fined-tuned STT model for Yoruba speech recognition, fine-tuned TTS model and the developed interface for Yoruba language speaker AI experience. The presentation begins with results of the fined-tuned STT model.

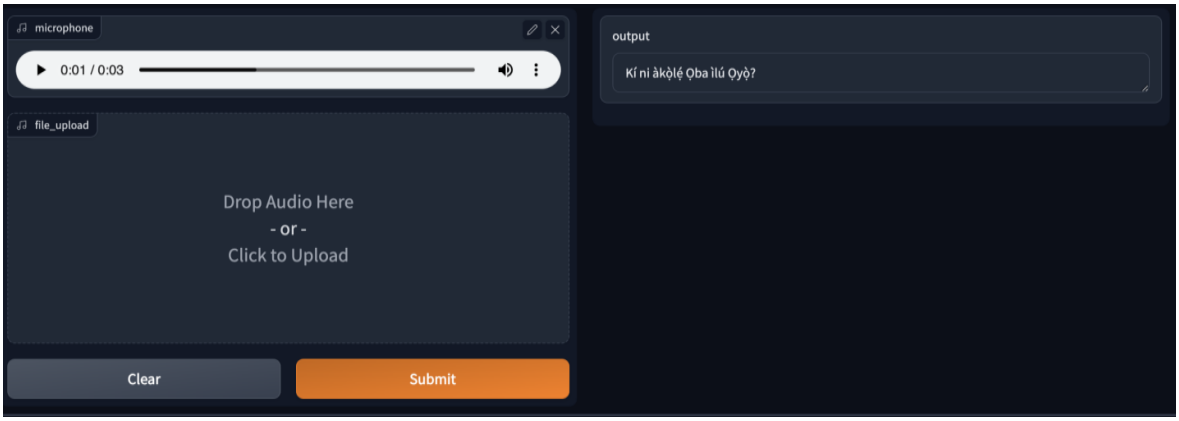
3.1 Fine-tuned STT model for Yoruba speech recognition

Putting the model into real-world test, some of the results achieved were shown Figure 3. The images consist of a simple interface with two sections, one section is for recording the speech using the microphone or dropping an audio file and the other section is for viewing the output of the transcribed Yoruba text. Figure 3 illustrates typical results obtained from fine-tuned STT model for automatic Yoruba speech recognition while Table 1 presents values of fine-tuned parameters of the STT.

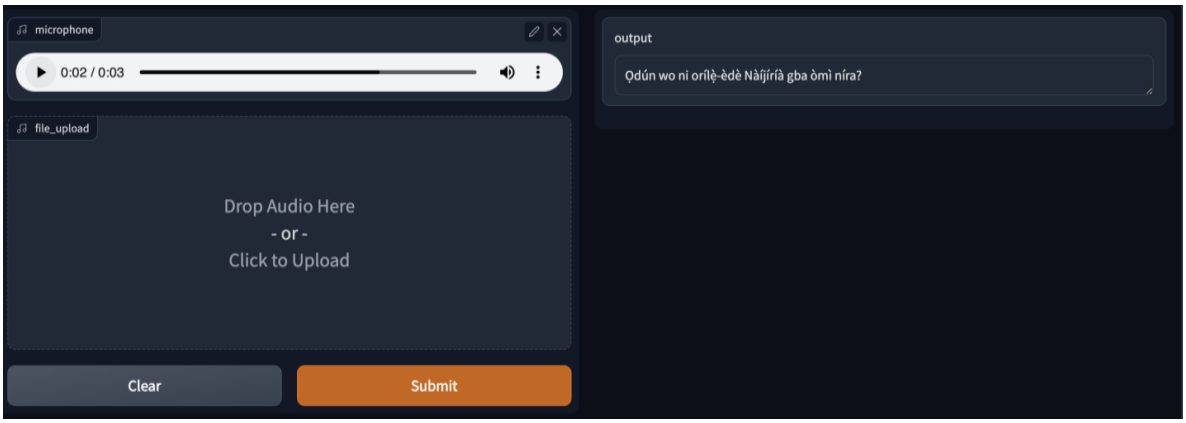
In each case of Figures 3(a)-3(c), the input is through audio recording. The audio files input in Figure 3(a) is, “Kí ni ìtumo ìgbéyàwó?” while that of Figure 3(b) goes thus, “Kí ni àkolé Oba ilú Qyò?” The question asked through the audio recording in Figure 3(c) is, “Odún wo ni orílè-èdè Nàijíríà gba òmì níra?” Going by what are displayed in the right side of the snapshots in Figures 3(a) – 3(c). The three audio files are correctly rendered into Yoruba texts.



(a)



(b)



(c)

Figure 3. Snapshots of obtained outputs from fine-tuned STT model for different Yoruba audio files input (a) Kí ni ìtumo ìgbéyàwó? (b) Kí ni àkolé Oba ilú Qyò? (c) Odún wo ni orílè-èdè Nàijíríà gba òmì níra?

Table 1. Statistics of fine-tuned STT model for automatic Yoruba speech recognition

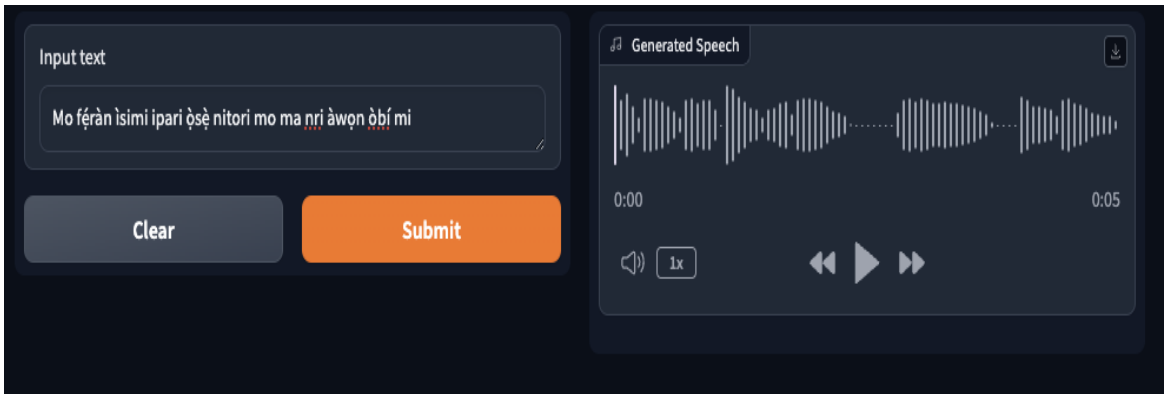
Parameters	Values
Training Loss	0.1267
Epoch	2
Validation Loss	0.1524
WER	13.0553%

Training a STT model involves monitoring key metrics like training loss, validation loss, and WER across epochs. Training loss shows how well the model is learning from the data while validation loss indicates its ability to generalize to new and unseen data. As the model trains over multiple epochs, it is expected that both losses should decrease, which should correspond to a lower WER, reflecting improved speech recognition accuracy. Together, these metrics help ensure the model is effectively learning without over fitting, resulting in better overall performance.

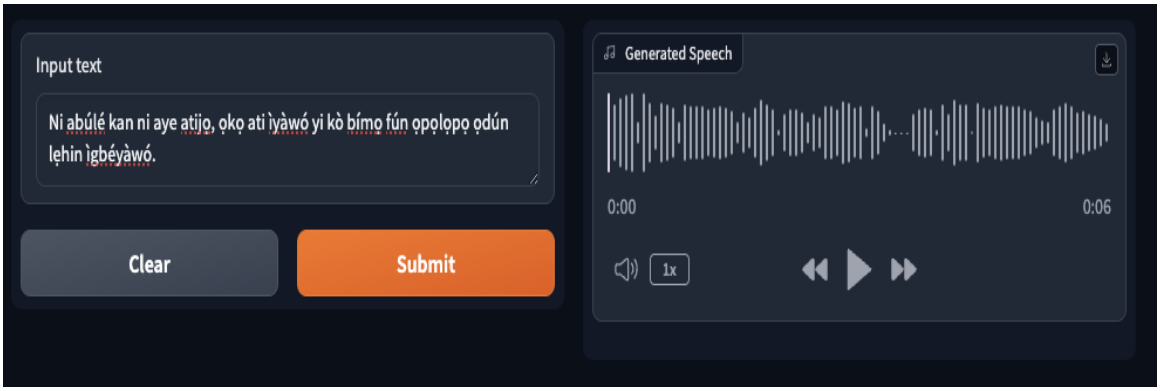
After training the STT model for approximately 1 hour on the GPU provided by Google Colab, the model was able to achieve a WER of about 13% which is the main metric used to evaluate the performance of the STT model. This means that 13% of the words generated in the transcribed text are incorrect. This include substitutions (where one word is replaced with another), insertions (where extra words are added), and deletions (where words are missing). This indicated that out of every 100 words in the transcribed text, about 87% are processed correctly.

3.2 Fine-tuned TTS model

Testing the model on some sample texts, the results together with the output spectrograms for two of the sample input texts were shown in Figure 4 while Table 2 presented associated statistics of the TTS model. During the training, the TTS model was optimized for the Character Error Rate (CER) between the predicted spectrogram values and the generated ones to minimize the difference between the predicted and the target spectrograms.



(a)



(b)

Figure 4. Snapshots of obtained output from fine-tuned TTS model when Yoruba text was fed into it (a) Mo féràn isimi ipari òsè nitori mo ma nri àwọn òbí mi (b) Ni abúlé kan ni aye atijo, oko ati lyàwó yi kò bímo fún opolopo odún lehin igbéyawó

Table 2. Statistics of fine-tuned TTS model for automatic Yoruba speech transcription

Parameters	Values
Training CER	17.0
Testing CER	16.1

Figure 4 illustrates what are obtained as outputs from two typical interactions with the YorubaAI system. In Figure 4(a), the Yoruba text input into the system is “Mo féràn isimi ipari òsè nitori mo ma nri àwọn òbí mi” while that of Figure 4(b) is “Ni abúlé kan ni aye atijo, oko ati lyàwó yi kò bímo fún opolopo odún lehin igbéyawó”. In each case of Figures 4(a) and 4(b), spectrogram of the input text are displayed at the right hand side of the interface.

Since TTS is a one-to-many mapping problem, i.e. the output spectrogram of a given text can be represented in many different ways. So this led to the use of a manual evaluation technique that allows human evaluators to rate the quality of the synthesized speech. Thus, qualitative assessment of the outputs the fine-tuned TTS is done through the mean opinion scores. This is a subjective scoring system that allows human evaluators mainly native speakers of the language to rate the perceived quality of synthesized speech on a scale from 1 to 5. These scores are typically gathered through listening tests, where human participants listen to and rate the synthesized speech samples. Ten native Yoruba speakers are involved in the ratings of synthesized speech. Five of the rated synthesized Yoruba language speech samples on a scale from 1 to 5 based on the perceived quality are:

Sample 1: Nítorí ọ̀rọ̀ táa sọ lójọ́sí ni Túndé yóò fi kúró nílé ìwé

Sample 2: Àwọn òṣìṣẹ́ ìjọba ti ń kọ́ àwọn èèyàn bí wọ́n ṣe lè dìbò

Sample 3: Iṣẹ́ yíyàwòrán ti di iṣẹ́ tí gbogbo ọ̀dọ́ fẹ́ ṣe.

Sample 4: Tani oludasilẹ Facebook?

Sample 5: Gbajúmọ̀ olukọ ni Fẹmi Ọlọ́runnísọla lásìkò ogun jálùmi

Table 3 presents results of mean opinion score of the fine-tuned TTS model on five different samples of Yoruba speech text by ten native Yoruba speakers.

Table 3. Mean opinion scores of fine-tuned TTS for ten users

Users	Speech Samples				
	1	2	3	4	5
1	3.5	4.0	4.0	2.0	2.5
2	3.0	4.0	4.2	2.0	3.0
3	3.0	3.5	4.0	1.5	2.5
4	3.2	3.7	4.0	2.0	2.7
5	3.0	4.0	3.5	2.0	2.3
6	3.5	4.0	4.5	2.5	3.0
7	2.9	3.2	4.0	2.0	3.0
8	3.7	3.9	4.2	1.7	2.5
9	3.0	4.0	4.0	2.0	3.0
10	3.0	3.5	4.5	2.5	2.5
Mean score	3.18	3.78	4.09	2.02	2.70

From the results shown in Table 3, it can be seen that the TTS model performed remarkably on the Yoruba language speech samples. This is obvious from the scores returned, especially for those speech samples that are purely in Yoruba language and tone marks are correctly assigned as seen in speech samples 1, 2 and 3. However, the performance of the model is not as satisfactory when speech samples are laced with incorrect tone marks as in sample 5 and when the speech contains words that are not Yoruba as can inferred from speech sample 4 that has English language word, “Facebook”.

3.3 Developed interface for Yoruba language speakers AI experience

Different building blocks (TTS, STT, RAG and LLM) are combined together using a simple interface for interaction. The interface is divided into two sections, one for the input and the other for the output. The input format can be a live audio recording using the microphone of the device on which the YorubaAI system is running on, or a pre-recorded audio which can be uploaded. The output section consists of the transcribed text of the audio question, the generated text answer and the spectrogram of the audio answer. Some of the results obtained when the YorubaAI system was put into test were shown in Figures 5 - 7.

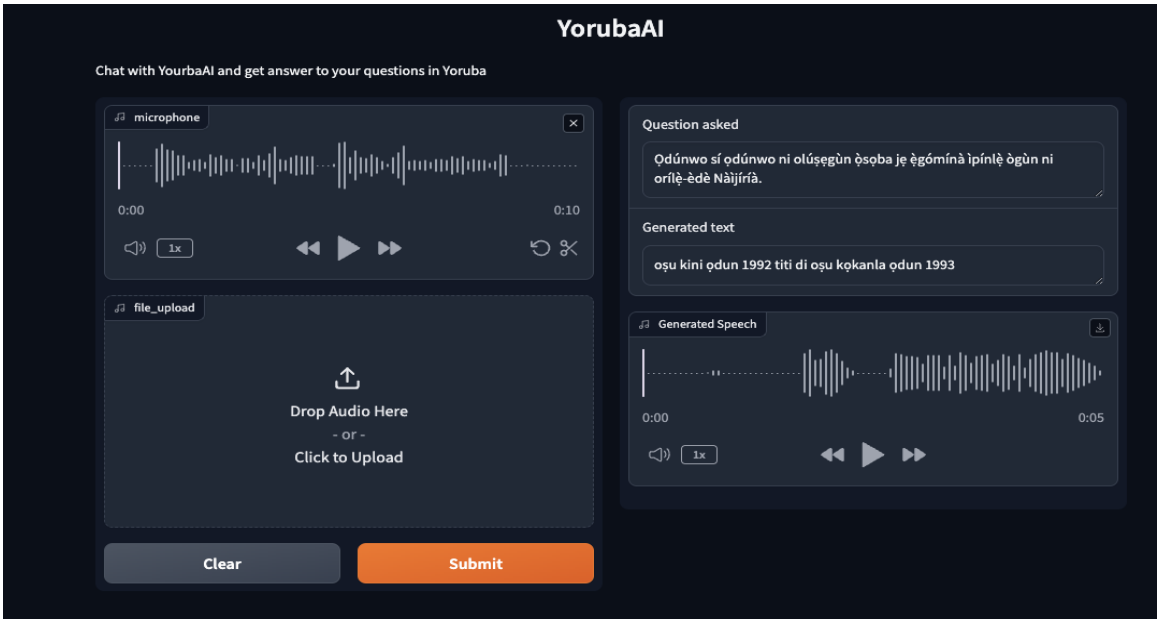


Figure 5. Snapshot of the composite output from the interface given input (Ọdúnwo sí ọdúnwo ni Olúṣẹgùn Ọ̀ṣọba jẹ ẹ̀gómínà ìpínlẹ̀ ògùn ni orílẹ̀-èdè Nàìjíríà)

Figure 5 depicted the situation when the input demanded for the period Olusegun Osoba was the Governor of Ogun State, Nigeria (Ọdún wo sí ọdún wo ni Olúṣẹgùn Ọ̀ṣọba jẹ gómínà ìpínlẹ̀ ògùn ni orílẹ̀-èdè Nàìjíríà). Based on the knowledge of the LLM and the available data given, YorubaAI was able to generate the correct response which is from January 1992 to November 1993 (oṣu kini ọdun 1992 titi di oṣu kọkanla ọdun 1993). The generated text was then converted to audio and the spectrogram of the generated audio was displayed in the last part of the output section.

Figure 6 showed the case when the YorubaAI system was asked a question that the LLM used do not have knowledge of, the system responded with “rara o” instead of trying to make up wrong answers and misled the user. This situation depicted the case when the database did not contained relevant data on question asked. In Figure 6, the system was asked about the first man to land on the moon (Ta ni ẹni tí òkọ́kọ́ dé orí oṣùpà?). Going by the response of the system, it showed that YorubaAI had no knowledge about the question, the output “rara o” was returned. Figure 7 portrayed what was obtained when the system was asked about the number of states in Nigeria (ipínle mélo ló wà ni Nàijirià), the system was able to use the knowledge of the LLM and the database to provide the correct answer which is 36 (merindinlogóji).

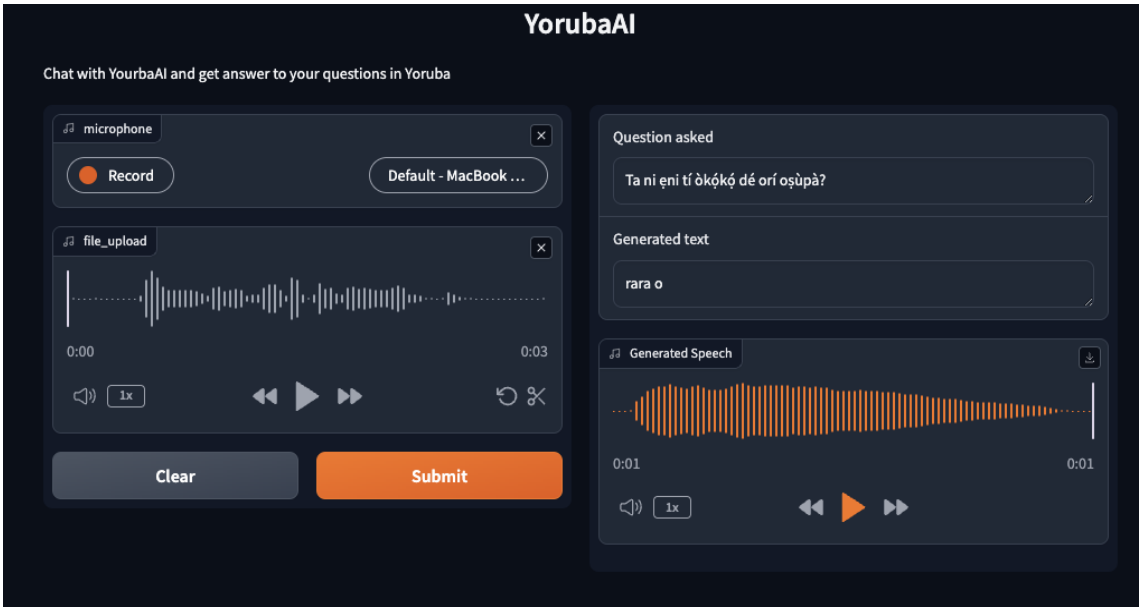


Figure 6. Snapshot of the composite output from the interface given input (Ta ni ẹni tí òkọ́kọ́ dé orí oṣùpà?)

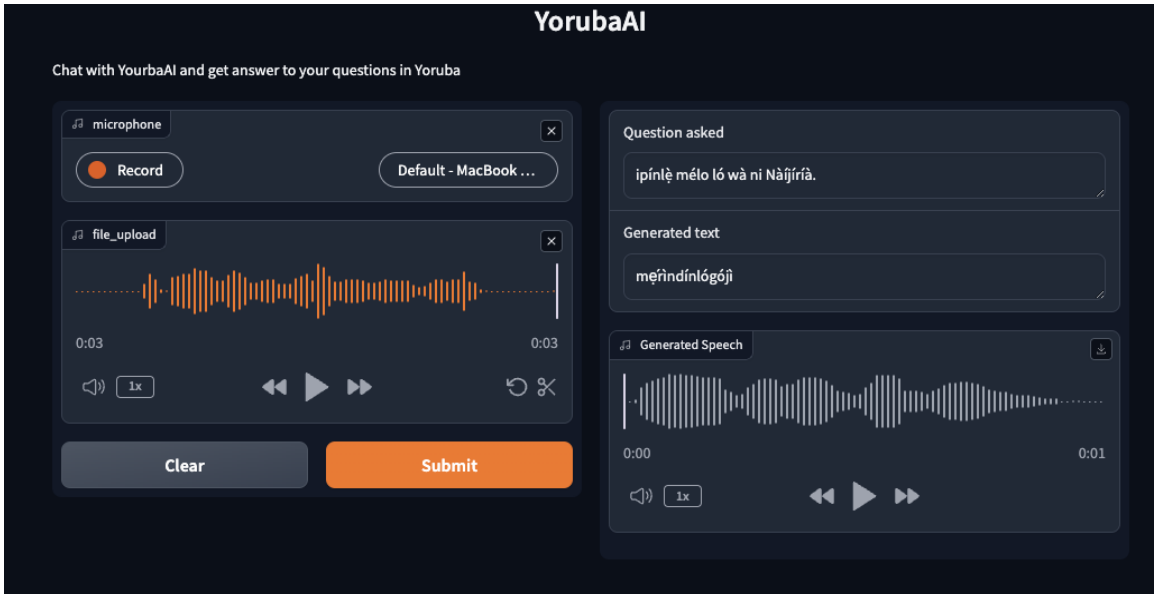


Figure 7. Snapshot of the composite output from the interface given input (ipínle mélo ló wà ni Nàijirià)

4. Conclusion

This study develops an AI assistant system, codenamed YorubaAI, which allows Yoruba language speakers to interact with advanced language models, such as ChatGPT and Bard, through Yoruba speech and text. In the development, the Whisper model was adapted for Yoruba speech recognition using audio samples from the Mozilla Common Voice 13 dataset. The model achieved a Word Error Rate of about 13%, indicating roughly 87% accurate transcription of Yoruba speech to text. A pre-trained TTS model from Facebook, already fine-tuned on Yoruba speech, was used to convert Yoruba language text to natural-sounding speech. The performance of the TTS model was evaluated through Character Error Rate and Mean Opinion Score ratings by native Yoruba language speakers. For character error rate, values of 17% and 16% were obtained for training and testing, respectively, while mean opinion score by 10 native speakers on a scale of 5.00 returned values between 3.18 and 4.09 for

pure Yoruba speech transcription. These results demonstrated the effectiveness of the TTS model for the assigned task.

Gradio framework was utilized to create a seamless interface, allowing users to input voice queries and receive responses in text and speech formats. The system combines STT, TTS, RAG and LLM components to handle queries accurately. The YorubaAI system integrates STT, TTS and LLM, providing an end-to-end AI experience in the Yoruba language. The RAG enhances the system's ability to fetch relevant data for accurate responses. YorubaAI promotes digital inclusivity by enabling millions of Yoruba language speakers to access AI technologies in their native language, supporting cultural preservation and expanding opportunities for education, business, and daily interactions. The study also demonstrates the potential for AI systems to support low resource languages through models' fine-tuning and adaptation.

Suggestions for future work

Expansion of the database will be looked into to increase the size of the Yoruba texts question-answer database to improve the system's accuracy and range. In addition, transitioning to LLM fine-tuning is another area of improvement. As the database grows, directly fine-tuning the LLM with Yoruba text data will enhance response generation and reduce reliance on retrieval techniques. Lastly, improving TTS for mixed-language texts is equally important. The current TTS model struggles with English words within Yoruba texts to improve pronunciation and naturalness.

References

[1] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang *et al.* "A survey on evaluation of large language models". *ACM Trans. Intel. Syst. Tech.*, vol. 15, issue 3, pp. 1-45, March 2024.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.* "Attention is all you need." *31st Conf. Neural Infor. Process. Syst. (NIPS 2017)*, pp. 6000-6010, Long Beach, CA, USA, Dec. 2017.

[3] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *In Proc. 2019 Conf. of the North American Chapter of the Assoc. for Computat. Linguist: Human Lang. Technol., Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang *et al.* "Exploring the limits of transfer learning with a unified text-to-text transformer". *J Mach. Learn. Res.*, vol. 21, no. 140, pp. 1-67, June 2020.

[5] OECD Digital Economy Paper. "AI language models: Technological, socio-economic and policy considerations". *OECD Publishing*, No. 352, pp. 1-52, April 2023. DOI: 10.1787/13d38f92-en

[6] C. Wei, Y.C. Wang, B. Wang, and C.C.J. Kuo. "An overview on language models: Recent developments and outlook". *APSIPA Trans. Signal Infor. Process.*, vol. 13, no. 2:e101, 1-49, Feb. 2024

[7] X. Lu, S. Li. and M. Fujimoto. "Automatic speech recognition". *In Book: Speech-to-Speech Translation*, Springer Singapore, 2020, pp. 21-38. DOI: 10. 1007/978-981-15-0595-9_2.

[8] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever. "Improving language understanding by generative pre-training". *OpenAI Pre-print*, pp. 1-12, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[9] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou *et al.* "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". *Proc. 2021 Conf. of North American Chapter of the Assoc. Comput. Ling.: Human Lang. Technol.*, pp. 483-498, June 2021.

[10] T. J. Sefara, S. G. Zwane, N. Gama, H. Sibisi, P. N. Senoamadi and V. Marivate. "Transformer-based machine translation for low-resourced languages embedded with language identification". *In 2021 Conf. Infor. Commun. Tech. & Soc. (ICTAS)*, Durban, pp. 127-132, 2021.

[11] F. Dhanani and M. Rafi. "Attention transformer model for translation of similar languages". *In Proc. Fifth Conf. Mach. Translat.*, Virtual, pp. 387-392, 2020.

[12] G. Anmol, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang *et al.* "Conformer: Convolution-augmented Transformer for Speech Recognition", *Proc. Interspeech*, Shangai, China, pp. 5036-5040, 2020

[13] M. N. Abdou, A. Allak, K. Gaanoun, B. Imade, Z. Erraji and A. Bahafid. "Multilingual speech recognition initiative for African languages". *Int J Data Sci Anal.*, 2024. doi:10.1007/s41060-024-00677-9

[14] S. Ralethe. "Adaptation of deep bidirectional transformers for Afrikaans language". *In Proc. 12th Lang. Res. Evaluat. Conf.*, Marseille, France, pp. 2475-2478, May 2020.

[15] R. Liu, Y. Hu, H. Zuo, Z. Luo, L. Wang and G. Goa. "Text-to-speech for low-resource agglutinative language with morphology-aware language model pre-training". *IEEE/ACM Trans. Aud., Speech, and Lang. Process.*, vol. 32, pp. 1075-1087, 2024.

[16] O. Isaac. "Machine Translation System for Numeral in English Text to Yorùbá Language". *Ife J. of Infor. Commun. Tech.*, vol. 6, pp. 26-37, 2022.

[17] C. F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le *et al.* "Transformer-transducer: End-to-end speech recognition with self-attention". *arXiv preprint*, 2019. *arXiv:1910.12977*

[18] N. Banar, W. Daelemans and M. Kestemont. "Character-level transformer-based neural machine translation". *In Proc. 4th Int. Conf. Nat. Lang. Process. & Infor. Retr.*, Soel, South Korea, pp. 149-156, 2020.

[19] C.B. Clement, D. Drain, J. Timcheck, A. Svyatkovskiy and N. Sundaresan. "PyMT5: multi-mode translation of natural language and Python code with transformers". *Proc. 2020 Conf. Empirical Methods in Nat. Lang. Process.*, pp. 9052–9065, Nov. 16–20, 2020.

[20] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova and M. Kudinov. "Grad-TTS: A diffusion probabilistic model for text-to-speech". *In Int. Conf. Mach. Learn.*, Virtual, pp. 8599-8608, July 2021.

[21] A. Radford, J.W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever. "Robust speech recognition via large-scale weak supervision". *In Int. Conf. Mach. Learn.*, Honolulu, Hawaii USA, pp. 28492-28518, July 2023.

[22] OpenAI. "GPT-4 Technical Report". *OpenAI*, pp. 1-100, 2023. https://cdn.openai.com/papers/gpt-4.pdf

[23] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna *et al.* "Google USM: Scaling automatic speech recognition beyond 100 languages", *arXiv preprint*, 2023. *arXiv:2303.01037*.

[24] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang *et al.* "Zipformer: A faster and better encoder for automatic speech recognition". *Proc. ICLR 2024*, pp. 1-16, 2024.

[25] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao *et al.* "Fastspeech: Fast, robust and controllable text to speech". *Proc. 33rd Int. Conf. Neural Infor. Process. Syst.*, vol. 285, pp. 3171-3180, 2019.

[26] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao *et al.* "Fastspeech 2: Fast and high-quality end-to-end text to speech", *arXiv preprint*, 2020. *arXiv:2006.04558*.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.* "RoBERTa: A Robustly Optimized BERT Pre-training Approach", 2019. http://arxiv.org/abs/1907.11692v1

[28] S. Gandhi, M. Hollemans, M. Khalusova, V. Srivastav. "HuggingFace Audio Course". https://huggingface.co/learn/audio-course/en/chapter5/introduction Accessed Feb. 16, 2024.

[29] Webmob Software Solutions (WSS). "How can NLP-based Language Translation solve Real-World Problems?" https://www.linkedin.com/pulse/how-can-nlp-based-language-translation-/ Accessed Feb. 16, 2024.

[30] Mozilla Discourse. "Common Voice Dataset 13", 2023. https://discourse.mozilla.org/t/dataset-13-release/112216

[31] F. Feng, Y. Yang, D. Cer, N. Arivazhagan and W. Wang. "Language-agnostic BERT Sentence Embedding". In *Proc. 60th Annual Meeting of the Assoc. Computat. Linguist. (Vol.1: Long Papers)*, Dublin, Ireland, pp. 878–891, 2022.