NEXT WORD PREDICTION IN YORÙBÁ TEXT: A LONG SHORT TERM MEMORY APPROACH

AYODEJI OBAUDE[1]
EZEKIEL OYEKANMI[1]
JOSHUA TOM[1]

Achievers University, Owo
Department of Computer Science
P.O. Box 1020, Km 1, Idasen-Ute Road, Owo, Ondo State, Nigeria
[1]obaude-pg0011@achievers.edu.ng, [2]e.oyekanmi@achievers.edu.ng, [3]tomjj@achievers.edu.ng

Abstract. Yorùbá is a language spoken by over 50 million people and being a tonal and diacritic language, it poses a significant challenge to users in the digital space. The lack of adequate language-smart data input systems and keyboards that support the language increases the hurdles of communicating in the language in the digital space; the lack of this basic language software text input has forced younger generations to transition from their mother tongue to a high-resource language that has massive software support, thereby leading to the wearing off of the Yorùbá cultural identity in the digital space gradually. This paper proposes the development of three LSTM next-word-based text prediction models for the Yorùbá language to enhance digital typing and promote linguistic accuracy on social media networks. The dataset for this study was obtained from the OpenSLR 86 public repository. This study implements and compares three deep learning models, namely the Vanilla LSTM, the Stacked LSTM, and the Attention LSTM model. The findings show that the Vanilla LSTM outperforms all other complex models, achieving the highest Tok-k accuracy and a lower perplexity of 165.90 as against other models with over 273.55 and 286.26 perplexities. These findings show that the word-based next-word prediction system developed in this study will not only preserve diacritical marks and reduce the cognitive load of Yorùbá speakers when typing but will also preserve Yorùbá culture across digital space.

Keywords: Natural Language, Low-resource, Yoruba

(Received June 20, 2026 / Accepted July 31, 2026)

1 Introduction

Natural language processing (NLP) has revolutionized how individuals communicate in digital spaces. Despite this, most digital software systems and tools developed are based on high resource language sources, such as Chinese and English [11]. Out of the 7,000 languages worldwide today, only a small fraction of these languages, especially English and Mandarin, dominate the digital space [9]. African languages are underrepresented, especially the Yorùbá language. This results in speakers of this language lacking fundamental tools that can aid their digital communication, such as autotext completion, which increases language use among the younger generation [8], [15] also highlighted this prevailing problem of limited research on speech recognition in the Yorùbá language, which was attributed to a lack of tools and tonal difficulties. This made it difficult for non-tonal models. Yorùbá is spoken by around 50 million people, but its tonal nature poses significant challenges for NLP tasks [13]. [1]noted that the standard Unicode system cannot represent some complex Yorùbá text in a single Unicode code point. These letters contain an underdot and a tone mark that require several character combinations to produce the de-

sired letter, which adds to the hurdles of typing Yorùbá words.

The standard keyboard does not easily represent Yorùbá tonality and diacritic properties. Morphologically, Yorùbá appears in abundance through agglutinative components (joins added to roots), compounding (combining words into new ones), and code-switching with English or Nigerian Pidgin in present-day usage. This introduces complex word arrangements as sentences mix one language with another. This leads to more challenging tokenization and modeling, thereby affecting the performance of the model . Diacritics and tones can introduce ambiguity because omitting them can turn distinct words into homographs, which confuses models trained without them [8]. For researchers in computational phonetics and African studies, this contributes to understanding how repetitive models adjust to tonal morphology and low-resource settings. For common readers, including Yorùbá speakers, students, teachers, and parents, it simply means hope: a better tool to type, write, read, and share in their language without constant errors or frustration. In a digital world ruled by a few languages, ensuring that Yorùbá flourishes preserves identity, history, and provides a sense of belonging for millions.

1.1 Problem statement

Difficulty in typing Yorùbá letters is one of the main problems in Yorùbá digital communication due to the lack of a smart, language-aware input tool. Because Yorùbá is a tonal and diacritical language that requires extra steps, many users switch to English due to the lack of tools and the complexity of the language. This slowly reduces the use of written Yorùbá, as noted by [13]. Without a text suggestion system, people often type words without tone marks due to haste. However, this creates confusion because readers struggle to understand the meaning from the context. In addition, the existing Unicode system does not have a single code point that represents all Yorùbá characters. This, therefore, makes correct typing more difficult, even when users try, as highlighted by [1]. Also, the existing dictionary-based prediction system does not understand grammar rules or sentence context. Therefore, there is a need for a smart, intelligent system that addresses this problem.

1.2 Research Aims and Objectives

The study aims to develop three LSTM-based next-word prediction models for the Yorùbá language to enhance digital typing and promote linguistic accuracy on social media networks. Research Objective In order to achieve this aim, the research follows the stated objective:

i) curate sets of Yorùbá texts from online repositories while retaining the diacritics.

ii) design a comprehensive system architecture for developing next word prediction model in Yorùbá

iii) implement the development a next-word prediction model for sequential processing of Yorùbá words.

iv) evaluate the performance of the model using various metrics such as accuracy and perplexity score in order to measure the fluency against human-generated output .

1.3 Contribution statement

There are several reasons this study contributes to knowledge. First, cultural preservation and linguistic vitality. One of the most significant contributions of this study is its aim to protect the Yorùbá language from fading in digital spaces. This is because most modern technology systems provide minimal support for low-resource languages like Yorùbá, leading to a reduction in writing proficiency in digital communication. In other words, this research promotes language preservation and accurate language use, especially among young people. Secondly, many African languages, such as Yorùbá, receive limited support from modern technology compared to high-resource languages like English, for which most technologies are designed. This research contributes to addressing this imbalance by providing a smart prediction system developed specifically for Yorùbá text while preserving contextual tone marks and meaning. It also aims to solve the challenges that make Yorùbá typing slow and difficult by developing a tool that enables Yorùbá users to use technology more easily and gives the language equal treatment in the digital world. Thirdly, traditional Yorùbá typing often requires multiple keystrokes to produce the desired Yorùbá letter or word. However, this can result in cognitive overload, leading to a trade-off between accuracy and speed. Through this research, the proposed model will reduce misinterpretation and improve readability across digital spaces. Lastly, the research offers methodological extensibility, as the framework can be tailored to other African languages with limited resources. The methodology of this study can serve as a benchmark for developing next-word prediction systems for languages such as Igbo, Hausa, Igala, and oth-

ers, thereby advancing wider multilingual and inclusive NLP research.

2 Literature Review

The origin of language modeling dates back to 1913, when Andrey Markov published the Markovian dependent probability, which was later expanded by his colleague, Shannon [16].[16] also highlighted that Markov proved that his model could be applied to language by analyzing the probability of the next vowel or consonant in the popular poem Pushkin's Eugene Onegin. Markov analyzed the poem by showing that the current state of a letter depends on its previous state.

Markovian dependent probability, one of the earliest methods employed to train models that predict the next word in a given sentence, relies on the N-gram as proposed by Markov in 1913. It works by predicting the next word, X_t, based on the probability of the word before it, $N-1$, as defined by the number of words to look back at.

$$P(w_t \mid w_{t-1}, w_{t-2}) = \frac{\text{count}(w_{t-2}, w_{t-1}, w_t)}{\text{count}(w_{t-2}, w_{t-1})} \quad (1)$$

Where P is the Probability of a word occurring given the previous words
W_t is the current word
W_{t-1} is the previous word before current word
N is the total number of word
$Count$ refers to the frequency of that particular word appearing relative to the total word.

Despite the elegance and simplicity of the N-gram model, one prominent issue with this method is its local dependency, which might lead to a loss of context, especially when a word relies on an earlier word that was not captured by the length of the gram to look back. This N-gram however served as the foundation for methods used in language models before the invention of neural-based large language models, as noted by [7].

The recurrent neural network introduced by David Rumelhart in 1986 for sequential modeling uses self-learning, which allows a neural network to organize its own internal representation, as noted by [4]. [10] noted that simple recurrent neural networks are powerful, but they often struggle with the vanishing gradient problem, where the network forgets the earlier words of a long statement because their gradient weights are lost or become insignificant during gradient descent. This specific issue marked the beginning of innovative approaches needed to prevent the vanishing gradient problem. One prominent solution to this is the introduction of the LSTM model, which is aimed at addressing the limitations of simple or vanilla RNNs through its gating properties. The LSTM introduced by Schmidhuber in 1997 provides a specific mechanism for how information is processed, namely, which information to store and discard, which helps prevent the loss of learning ability. As identified by [12], a model's learning ability is tied to its gradient, which is the derivative of the weight and its bias. The LSTM protects this learning ability through its gating mechanism . This is important for modeling Yorùbá next-word prediction, as it allows the model to reserve early tonal and word information to ensure the final predicted word is accurate.

The LSTM internal state is governed by :

a) Forget gate: This gate(F_t) determines, based on the previous cell, which information is no longer needed and will be discarded. It is mathematically represented as :

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

Where :
F_t is the forget gate vector (values between 0 and 1) as determined by the sigmoid function.
σ is the sigmoid activation function
W_f is Weight matrix for the forget gate
h_{t-1} is the previous hidden state
x_t is the current input
b_f is the bias or intercept of the forget vector

b) Input gate : The input gate (I_t) is responsible for determining what information will be kept in the long memory cell The input gate is represented mathematically as :

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

Where :
I_t is the input gate vector (values between 0 and 1) as determined by the sigmoid function.
σ is the sigmoid activation function
W_i is Weight matrix for the input gate
h_{t-1} is the previous hidden state
x_t is the current input
b_i is the bias or intercept of the input gate

c) Candidate State : The candidate cell state determines the new information that the LSTM creates at the current time step, which is to be added to the cell state memory.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

Where :
$\tilde{C}_t$ represent the candidate cell state
$\tanh$ is the hyperbolic tangent activation function
W_c is Weight for the candidate cell

h_{t-1} represent the previous hidden state
x_t represent the current input
b_c represent the bias or intercept of the candidate cell

d) Cell state: The cell state (C_t) represents the actual memory cell, where information is stored or discarded. The Cell state is represented mathematically as :

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{5}$$

Where :
C_t is the updated current state
F_t is the forget gate
C_{t-1} is the cell state of previous time step
I_t is the input gate
$\tilde{C}_t$ is the candidate cell state
$\odot$ represent element wise multiplication

e) Output gate: The Output gate (O_t) determines the retrieval mechanism for the LSTM model; the output gate is responsible for providing output usable by users. The output gate is represented mathematically as :

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

Where:
O_t is the output gate
σ is the sigmoid activation function
W_o is Weight matrix for the output gate
h_{t-1} is the previous hidden state
x_t is the current input
b_o is the bias or intercept of the output gate

f) Hidden State : The hidden state represents the short term memory of the LSTM. For understanding sequential data

$$h_t = O_t \odot \tanh(C_t) \tag{7}$$

Where:
h_t represent the hidden state
O_t represent the output gate
$\tanh$ represent the hyperbolic tangent activation function
C_t represent the current cell state

2.1 Related Work

While standard LSTMs provide a framework for sequential modeling of words, recent studies have employed various types of this framework for low-resource African languages. [2] identified challenges in modeling Yorùbá text and noted the issue of having a small training and testing dataset. The study used the Byte-Level Byte Pair Encoding approach to tokenize the corpus and trained the model with a Bi-LSTM, a variant of the LSTM family that considers words from both directions and is commonly used for text generation. The model performed well on five of ten references, achieving a BLEU score of 1. While this approach addresses the limitations of small datasets, it results in high computational resource usage due to the Byte-Level Byte Pair Encoding nature of word breaking. Though the computational resources used are less expensive than those for character-level vocabularies, they are still more expensive than those for simple word tokens.

Another study by [14] introduced an attention-augmented recurrent neural network to address information bottlenecks in RNNs. This mechanism outperformed the baseline model, achieving a perplexity of 2.21 compared to the baseline of 12.67. One primary problem with this approach is that it is computationally expensive due to the mechanism it uses. It compares the token to each word in the sequence [14] also found that a two-layer stacked LSTM reduces perplexity by 33.6% compared to a single LSTM layer. [17] employed a CNN and BiLSTM architecture for next-word prediction in bilingual social media text. The study achieved a training accuracy of 0.9367, showing the feasibility of developing next-word prediction for complex language arrangements such as multilingual text. [5] developed a BLSTM-GRU model for next-word prediction in Amharic. The model achieved an accuracy of 78.6 percent with a data corpus containing 63,300 Amharic sentences, outperforming LSTM, GRU, and Bi-LSTM models. The results of these findings facilitate digital inclusion for the language.

3 Research Methodology

This research study employs a quantitative experimental research design aimed at developing a next-word prediction model using three variant Long Short-Term Memory (LSTM) architectures for the Yorùbá language and evaluating the performance of the models using metrics such as Top-K accuracy and perplexity score. The research follows a systematic approach to developing the system, and the steps include the following.

3.1 System Architecture

This system architecture provides a comprehensive blueprint and process for the system design used to develop the next-word prediction model for the Yorùbá language as shown in Figure 1. The architecture of this study is engineered to preserve the diacritic marks in the Yorùbá language. The components of the system architecture are as follows.

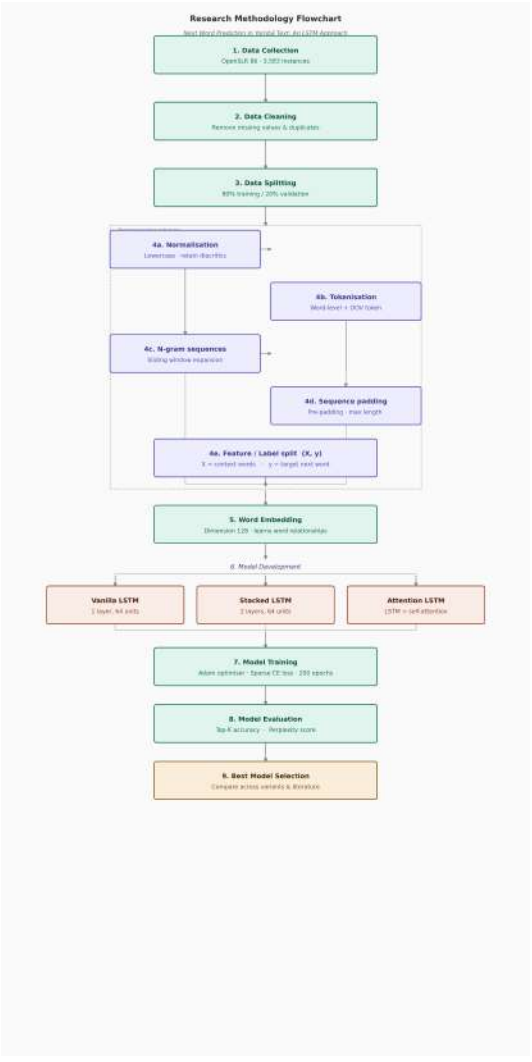

Figure 1: system architecture for the development of the LSTM-based next-word prediction model for Yorùbá.

3.2 Data Collection

The dataset used for this study was obtained from the OpenSLR 86 repository, which contains transcriptions of audio recordings from both female and male speakers. The dataset is arranged in a table that includes the feature names, the index of each data instance, and the transcribed words while maintaining the diacritics and undertone mark.

3.3 Data cleaning

The dataset underwent four data quality checks. Missing records were examined, and none were detected. Secondly, empty strings were checked, and none were detected. 560 duplicates were found in the sentence columns and were removed. After the removal of duplicates,1,121 disfluency annotation markers were detected and removed. Following this data cleaning procedure, the dataset was reduced from 3,583 to 3,023 unique sentences.

3.4 Data Preprocessing

Developing an LSTM model for next-word prediction in Yorùbá requires clean data and proper preprocessing to ensure that the data is ready for analysis and modeling. Poor-quality data will result in a poor model, and a poor model will lead to failed utilization in production, which may frustrate and discourage users. To ensure that the model is trained with a suitable dataset, the dataset underwent several preprocessing steps. These include:

3.5 Tokenization & Vocabulary Building

Most machine learning models cannot process raw text directly. Therefore, the sentences were broken down into tokens using the Keras Tokenizer, and the vocabulary mapped each tokenized word to an index, which was used for text embedding to establish relationships among the tokens.

3.6 N-Gram Sequence Generation and sequence padding

The tokenized dataset was sliced to create a sliding window for the next-word prediction model. The LSTM model requires a fixed-length vector as input. However, slicing the data into sequences results in variable-length data points. To achieve equal-length sequences, artificial entries were added through padding. Pre-padding was used to ensure that the target word remained at the end of each sequence, which is essential for next-word prediction using the LSTM model.

3.7 Embedding model

The embedding model establishes relationships among the tokens. In this study, the embedding model was adopted to give meaning to the data, which was intended to improve the model's performance and generalization. The embedding dimension used in this study was 128.

3.8 Model selection

Each machine learning or deep learning model has its strengths and weaknesses. For this study, in order to develop a model that is able to generalize, three variants of LSTM models were explored and adopted, namely the Vanilla LSTM, which is the basic form of the LSTM architecture; the Stacked LSTM, which employs two LSTM layers. The stacked model was employed because of the complexity of Yorùbá text as a diacritical and tonal language, which might be difficult for a Vanilla LSTM to capture. Finally, the Attention LSTM model was employed, as attention is an important component of modern deep learning models. The attention mechanism enables the model to identify the values or features that are most critical, allowing the model to better understand the context of the data, in this case, the Yorùbá language context.

3.9 Model Training and Model Evaluation

The model was trained using the various LSTM model variants together with the Adam optimizer. The model's performance was evaluated on the validation set using various metrics, including Top-K accuracy. Top-K accuracy indicates whether the actual word exists among the predicted words based on the value of K. For example, Top-3 accuracy represents the three most probable words predicted by the softmax function. If the actual word is "lo" and the top three predicted words are "lo", "ja", and "ma", then "lo" exists among the predicted words, making the prediction accurate. In this study, the model's performance was evaluated using Top-3, Top-4, and Top-5 accuracy. The model was also evaluated using perplexity score. A low perplexity score implies that the model is less confused when predicting words, whereas a high perplexity score implies that the model is more confused.

3.10 Comparison

The model performances were compared, and the model with the lowest error rate was selected as the best-performing model based on the Top-K accuracy score and perplexity score. Finally, the best performing model was compared with other word-based models reported in the literature.

4 Results

4.1 Dataset Acquisition

The dataset for this study was obtained from the Open-Source Corpus of Yorùbá Speech (SLR86), crowd-sourced and curated by Gutkin et al. [6]. Comprising 1,892 dataset instances for females and 1,691 dataset instances for males as shown in Figure 2.

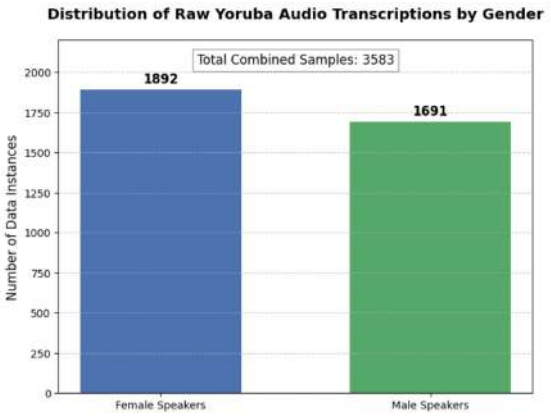

Figure 2: Distribution of the 3,583 audio-transcribed data instances in the Yorùbá corpus by speaker gender

4.2 Exploratory Data Analysis

In order to understand the characteristics of the data and also identify the best approach to be used that efficiently models the relationships in the dataset, several analyses were conducted to derive insights that can help generalize the behaviour of the model on unseen data. The insights derived are as follows :

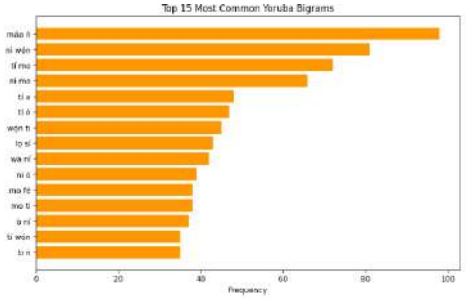

Figure 3: Top 15 most frequent word pairs.

The bigram analysis was conducted to identify the most common two words that occur together. This establishes that there is a strong local dependency in the Yorùbá corpus, as shown in Figure 3. The frequency of phrases such as 'máa n´' and 'ni wn' provides our LSTM with the possibility of statistical insight that allows the model to establish relationships in the dataset and infer phrases during production. This local dependency is what the LSTM's gates (Input and Forget gates) focus on to maintain short-term context during the prediction of the next word.

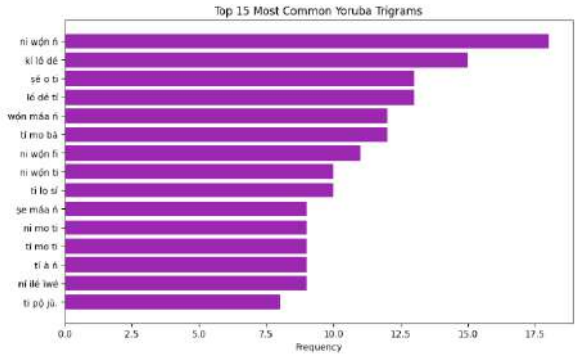

Figure 4: Top 15 most frequent three-word phrases

Figure 4 shows the top 15 three-word phrases that allow us to see how people speak most often. For example, 'ni wn ń' is the most common way people start a description. Figure 4 proves that the model will not only learn single words, but will also learn Yorùbá grammar patterns and common questions, which, in the end, make the next word guesses sound much more human rather than contextless and generic.

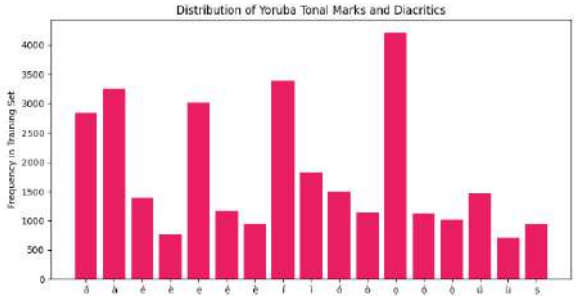

Figure 5: frequency distribution of each diacritic and tonal marker in the processed Yoruba Corpus.

As shown in Figure 5, the retention of the diacritic and tonal marks on Yorùbá processed words with a high frequency of '' and 'í' aligns with standard Yorùbá phonological patterns. This ensures that the model learns the distinction in the meaning of Yorùbá words and language. Higher data integrity will lead to more accurate data and enable the next-word prediction model to distinguish between words with similar spelling but different tonal marks.

4.3 Model Development and Architecture Selection

In order to determine the best model that is suitable for next-word prediction in Yorùbá text, this study developed three different architectural configurations based on the Long Short-Term Memory (LSTM) framework, starting from the baseline model, which is the Vanilla LSTM, then moving to the complex models, which are the stacked LSTM and the attention LSTM.

4.4 Experimental Configuration and Hyperparameters

The models developed in this study were implemented using TensorFlow and Keras in Python with an NVIDIA T4 GPU, which was provided by Google Colab. In order to ensure fairness, all models were trained using the same set of hyperparameters, as shown in Table 1.

Table 1: Model Hyperparameters

Hyperparameter	Value	Rationale
Optimizer	Adam	Adaptive learning.
Learning Rate	0.0005	Prevents overfitting.
Loss Function	Sparse Cat. CE	Multi-class classification.
Batch Size	32	Optimizes GPU.
Epochs	200	Ensures convergence.
Activation	Softmax	Multi-class output.

4.5 Vanilla LSTM Model

The vanilla model serves as the simplest model, which has less depth compared with subsequent models to be developed; it consists of an Embedding Layer of 128 dimensions, followed by an LSTM layer with 64 units as shown in Table 2. The dropout layer of 0.2 was integrated. The aim of this is to prevent overfitting so that the model can be generalized.

4.6 Stacked LSTM Model

The second model developed is the stacked LSTM model, an approach where two LSTM models are on top

Table 2: the architecture of Vanilla LSTM Model

Layer Name	Layer Type	Output Shape	Param #
Input	InputLayer	(None, 21)	0
Embedding	Embedding	(None, 21, 128)	395,648
LSTM	LSTM	(None, 64)	49,408
Dropout	Dropout (20%)	(None, 64)	0
Dense	Dense (Output)	(None, 3,091)	200,915
Total	**645,971 Params**	**2.46 MB**	

of each other. with the first layer (64 units) which return sequence of fixed length to the second layer which is also 64 units as shown in Table 3, the configuration allows the model to learn complex and in-depth grammatical features in the Yorùbá corpus .

Table 3: the architecture of Stacked LSTM Model

Layer Name	Layer Type	Output Shape	Param #
Input	InputLayer	(None, 21)	0
Embedding	Embedding	(None, 21, 128)	395,648
LSTM_1	LSTM (Seq)	(None, 21, 64)	49,408
LSTM_2	LSTM	(None, 64)	33,024
Dropout	Dropout (10%)	(None, 64)	0
Dense	Dense (Output)	(None, 3,091)	200,915
Total	**678,995 Params**	**2.59 MB**	

4.7 Attention-Based LSTM Model

Table 4 shows the attention LSTM model; it incorporates the self-attention mechanism between two LSTM layers, with the attention mechanism assigning weight to critical values in the sentence. This allows the model to focus on value which helps the model generalize and reduce loss of context.

Table 4: Architecture of Attention-Based LSTM Model

Layer	Type	Output	Param
Input	InputLayer	(None, 21)	0
Embedding	Embedding	(None, 21, 128)	395,648
LSTM_Enc	LSTM (Encoder)	(None, 21, 64)	49,408
Attention	Self-Attention	(None, 21, 64)	0
LSTM_Dec	LSTM (Decoder)	(None, 64)	33,024
Dropout	Dropout (10%)	(None, 64)	0
Dense	Dense (Output)	(None, 3,091)	200,915
Total	**678,995**	**Params**	**2.59 MB**

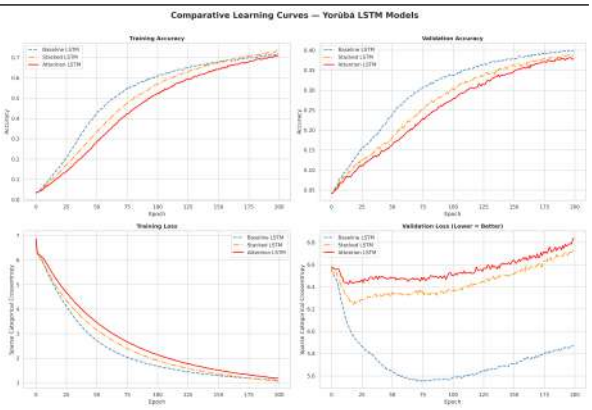

Figure 6: training /validation accuracy and loss for the vanilla , stacked and attention LSTM models for 200 epochs.

As shown in Figure 6, among the several algorithms used to develop the model, the vanilla LSTM model outperforms even the most complex models, namely the stacked LSTM model and the attention-based LSTM model; the vanilla LSTM has the lowest training loss along with the lowest validation loss. This shows that the basic model can better learn the patterns in the dataset. One main reason for this is that powerful models such as the stacked LSTM and attention LSTM require large amounts of data, which is not available in the study. On larger datasets, the Attention LSTM and the stacked LSTM model might outperform the vanilla LSTM.

4.8 Model Result and Evaluation

The model performance was evaluated using various metrics

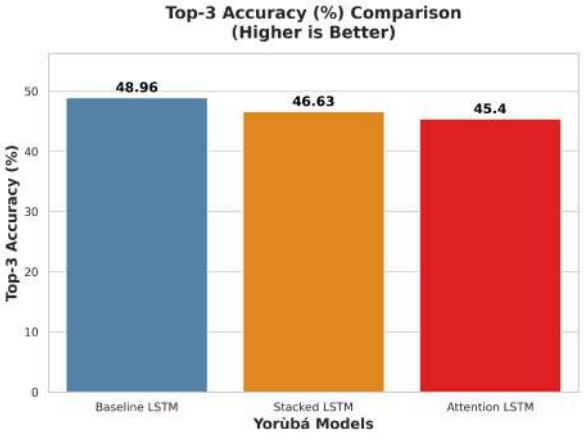

Figure 7: top 3 accuracy comparisons

As shown in Figure 7 the vanilla LSTM for its simplicity outperforms other models having the highest accuracy in top 3 word prediction with an accuracy of 48.96% followed by the stacked model achieving top 3 accuracy of 46.63% and lastly the attention LSTM

achieving a low accuracy of 45.4% . The decrease in error on both training and validation data and the top 3 accuracy confirms that the baseline model which is the vanilla LSTM outperforms other powerful models.

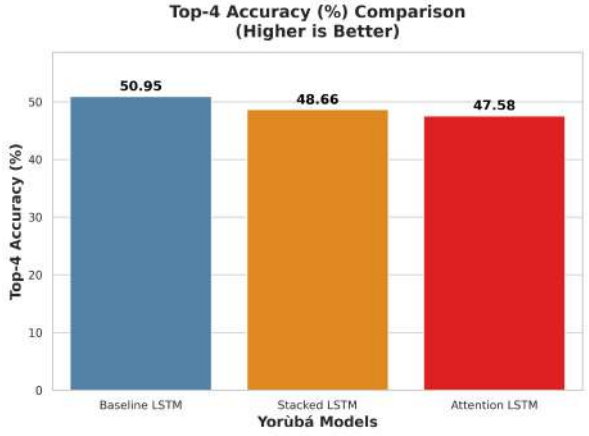

Figure 8: top 4 accuracy comparisons.

Similarly, as shown by Figure 8 the baseline LSTM also outperforms other models by achieving a top 4 accuracy of 50.95% followed by the stacked LSTM which achieves top 4 accuracy of 48.66% and lastly the attention LSTM achieving a top 4 accuracy of 47.58%. This shows that data scarcity plays a significant role due to the deteriorating performance of the high in-depth model.

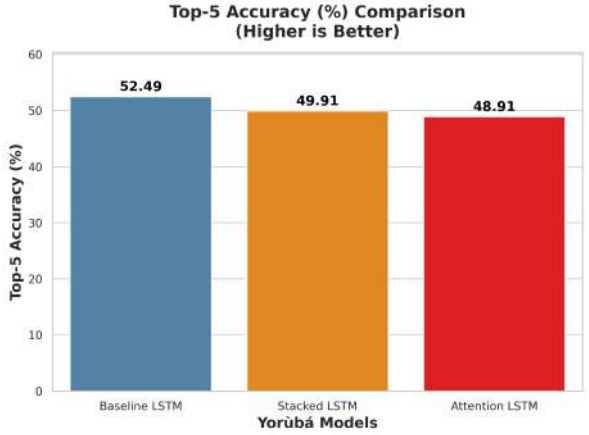

Figure 9: Top-5 accuracy comparisons..

The top 5 accuracy chart, as shown in Figure 9, also shows that the vanilla LSTM model outperforms other powerful models with a high score of 52.49% , while the stacked model has a top 5 accuracy of 49.91% and lastly, the attention LSTM achieves a low top 5 score of 48.91%.

Table 5: Perplexity Comparison Between LSTM Model Variants Developed For Next Word Prediction In Yorùbá

Model Architecture	Perplexity Score
Baseline LSTM	165.90
Stacked LSTM	273.55
Attention LSTM	286.26

The vanilla LSTM also achieved the lowest perplexity score of 165.90, which shows that it exhibits higher certainty when predicting the next word on the validation set as compared with other models. The stacked and the attention models achieved high perplexity scores of 273.55 and 286.26, respectively, as shown in Table 5. This implies these models, despite being powerful models, struggle to capture relationships and patterns in the dataset due to the small dataset size.

4.9 Statistical Significance Testing

The Cochran-Q test was used to check if there is a significant difference between the three models and the difference is not due to random chance, while the McNemar test was then used to compare pairs of models. Due to the comparison which involves three models, the Bonferroni correction was used, which set the level of significance at 0.0167. As shown in Table 6, Cochran's Q test shows that there is a significant difference between the three models. In addition, McNemar's test also shows that, in a situation where the model gave different answers, the Vanilla LSTM was more correct compared with the other two complex models, outperforming the stacked and the attention LSTM.

Table 6: Pairwise McNemar Test Results with Bonferroni Correction

Comparison	χ^2	p-value	Significant
Vanilla vs Stacked	9.47	0.002	Yes
Vanilla vs Attention	19.37	<0.001	Yes
Stacked vs Attention	2.01	0.156	No

4.10 Ablation Study

The study experimented with two ablations, which were conducted on the vanilla LSTM to measure how individual components contribute to the model performance. The dropout layer was removed for the first ablation, as shown in Figure 10 and Table 7; this reduced

the top-k accuracy across all levels and also increased the perplexity score from 165.90 to 410.78, which is a 148% increase. This confirms that dropout plays a vital role by contributing to the generalization of the model.

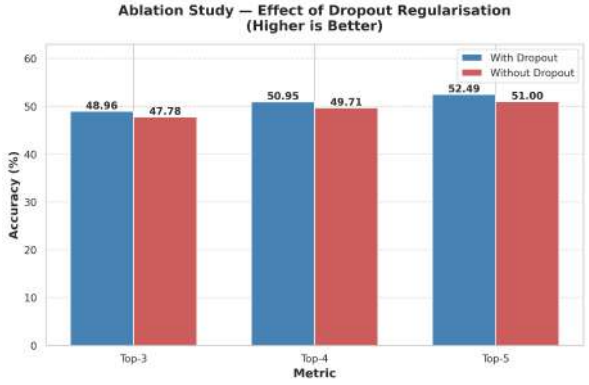

Figure 10: Top-K accuracy comparison between the vanilla LSTM with dropout and without dropout.

Table 7: Ablation Study on Dropout Regularisation Contribution

Metric	With Dropout	Without Dropout	Difference
Top 3 Accuracy (%)	48.96	47.78	−1.18
Top 4 Accuracy (%)	50.95	49.71	−1.25
Top 5 Accuracy (%)	52.49	51.00	−1.49
Perplexity	165.90	410.78	+244.88

Table 8: Ablation Study on the Contribution of Diacritic Preservation

Metric	With Diacritics	Without Diacritics	Difference
Vocabulary Size	3,091	2,358	−733
Top-3 Accuracy (%)	48.96	50.77	+1.81
Top-4 Accuracy (%)	50.95	53.80	+2.85
Top-5 Accuracy (%)	52.49	56.16	+3.67
Perplexity	165.90	84.96	−80.94

Table 8 shows the second ablation: diacritics and tonal marks were removed before tokenization; due to homographs, this reduced the vocabulary from 3,091 to 2,358 words. The performance of the model without diacritics increases with Top-5 accuracy of 56.16 percent and a low perplexity of 84.96. This is because the distinction of the words has been reduced significantly, which makes the task easier rather than the model improving. The homograph words are different Yorùbá words that exhibit different meanings, but the same spelling in the absence of diacritics and tone marks, and the model cannot distinguish the meaning of those words.

4.11 Error Analysis

An analysis was implemented using the validation set to determine where the model prediction was wrong in cases where the correct word did not appear in the Top 3 predictions, as described in Table 9.

Table 9: Prediction Failures from the Vanilla LSTM

Context	True Word	Top 3 Predicted	Failure Type
ó ti tóbi	jù	ju, ilé, bí	confusion due to diacritics
ti tóbi jù mí	ò	nínú, o, bí	High confidence error
níf láti máa j	edé	àkàrà, búrdì, dún	Semantic near miss
ní ni mo sùn	OOV	fún, l, táa	Out of vocabulary
orí ní ni	mo	wn, ibùs, àkànbí	Named entity

4.12 User Evaluation

A user evaluation was conducted among twelve Yorùbá speakers on the efficiency of the system. A web-based interface for develop for the system as shown in Figure 11. The system scored an average user rating of 4.5 on the overall performance of the model. One of the users who typed "mo ń b sí ilé" noted that after typing the first word, which was "mo," the entire word was predicted except "ń". Despite this, the user noted that the models prediction is still made sense.

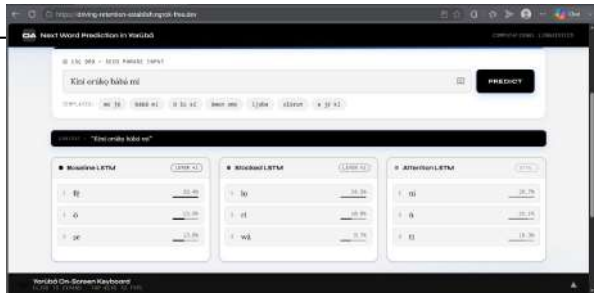

Figure 11: Yorùbá next-word prediction system interface showing the top 3 prediction for each LSTM models.

5 Discussion

This section presents the findings of our study on developing an LSTM-based next-word prediction model for the Yorùbá language, a system that accurately suggests words that sync with user preferences. It will not only reduce mental stress for users but will also ensure the inclusion of Yorùbá in modern technology, and this

Table 10: Performance Comparison of Model Strategies and Tokenization Units Across Model in Literature

Model Strategy	Tokenization Unit	Vocab Size (V)	Avg. PP	NPR (%)
[14] Attention RNN	Character based	25	2.21	8.84%
[2] Bi-LSTM	Sub-word (BPE)	500	48.50	9.70%
Current Study Vanilla LSTM	Word	3,091	165.90	5.37%

system will also ensure Yorùbá language and culture are well preserved among younger generations. This chapter also presents the comparison results among the three models developed and compares the performance of this model with models in the literature that are in a similar context.

The development of an LSTM-based next word prediction in Yorùbá language requires a well refined approach, to attain this aim the research followed the stated objective. The study also employed several preprocessing techniques to ensure the integration of the model goes seamlessly, techniques such as tokenization of text while addressing the unknown or missing vocabulary which might not be in the training data, arranging and padding. These ensure the data is well suitable for model development, embedding of the text and ensuring the model understands the context and relationship in the data.

5.1 Summary of findings

5.2 Model performance

Among the three models developed, the vanilla LSTM model shows an outstanding performance as compared to other complex models such as the stacked LSTM and the attention LSTM. The vanilla LSTM model achieves a high score from Top 3 accuracy score to Top 5 accuracy score and also achieves a perplexity score of 165.90 as compared to other models with perplexity of 273.55 and 286.26.

5.3 Dataset Size Limitations

The primary reason for the poor performance of the complex models compared with vanilla LSTM model is due to the small dataset which is in line with [3] findings that a less complex model performs well on a small dataset when compared to a complex model. According to [3], complex models exhibit more parameters, leading to overfitting during testing. In order to increase the performance of the complex models, an increase in the dataset size would be a better fit, and this is in alignment with [3] findings of using data augmentation for more complex models.

5.4 Comparison with existing models in literature

The model with the best performance developed in this study (Vanilla LSTM) was compared with existing models in the literature, as shown in Table 10. The Vanilla LSTM developed in this study achieves high perplexity of 165.90, as compared with [14] Attention RNN achieving a perplexity of 2.21 and [2] Bi-LSTM achieving a perplexity of 48.50 .

5.5 Limitation of model comparison across the literature due to different levels of tokenization

Our approach employs a word-based tokenizer instead of a character-based tokenizer, which might lead to loss of grammatical rules and an increase in overhead computation cost. Comparing a word-tokenized model with a character-based model by [14] using perplexity is inappropriate, as perplexity is directly tied to vocabulary length. A character-based Yorùbá vocabulary has 25 vocabulary items if we consider all the letters in the Yorùbá alphabet. In order to create a balanced comparison, we consider using the ratio of the perplexity with respect to vocabulary length.

6 Conclusion

This study successfully developed a robust next-word prediction system for the Yorùbá language using the Long Short-Term Memory (LSTM) architecture through rigorous procedures and data preprocessing techniques, considering the complexity of the language as a low-resource tonal language. This study proved that complex machine learning and deep learning algorithms are not, by default, better than simpler deep learning algorithms in every use case. The Vanilla LSTM model outperformed even the most complex architectures, achieving the lowest perplexity score, the highest Top-3 to Top-5 accuracy scores, and the lowest Normalized Perplexity Ratio of 5.37% as described in Table 10, which is superior to the character-level model reported in the literature. Finally, this study provides a technical and mathematical foundation for the preservation of the Yorùbá language in the digital space. The study also proved that, with the right preprocessing techniques, especially those aimed at preserving diacritic marks, the developed model can reduce the cognitive load on Yorùbá speakers and writers, allowing them to communicate more easily and fluently in their mother

tongue while using modern technology and digital platforms.

6.1 Future Research

While this study successfully made it possible to develop a next-word prediction model for Yorùbá, it also identified areas in which future researchers can explore more, most especially with the poor performance of the complex models.This poses a problem that needs to be addressed which is the vanilla model outperforms both the stacked and the attention LSTM models. In order to build on these findings and to give answers to the questions posed by this study, the following suggestions are proposed for future research.

Future researchers can make use of large datasets; this will enable complex models to fully leverage their high hyperparameter counts and allow them to learn complex patterns in the dataset and achieve superior performance that can generalize more to unseen data without the risk of overfitting.

Future researchers can make use of transformer models to develop a next-word prediction system for Yorùbá text. To mitigate the problem of data scarcity, Future researchers can employ multilingual pretrained models such as BERT and RoBERTa and fine-tune these models for Yorùbá use cases; in this way, the model can generalize more.

Funding

There was no funding from any organisation of any kind in this study.

Conflict of Interest

There was no conflict of interest of any kind between the authors and any organisation.

References

[1] Ajao, J. F., Babatunde, R. S., Asapetu, S. O., and Yusuff, S. R. Implementation of Yorùbá unicode generation for an indigenous keyboard. *Technoscience Journal for Community Development in Africa*, 1(1):71–80, 2020.

[2] Aliyu, E. O. A deep learning approach for Yoruba language next word generation. In *2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON)*, pages 1–4. IEEE, 2024.

[3] Brigato, L. and Iocchi, L. A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2490–2497. IEEE, 2021.

[4] Das, M. and Alphonse, P. J. A. A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset. *arXiv preprint arXiv:2308.04037*, 2023.

[5] Endalie, D., Haile, G., and Taye, W. Bidirectional long short term memory-gated recurrent unit model for Amharic next word prediction. *PLOS ONE*, 17(8):e0273156, 2022.

[6] Gutkin, A., Demirşahin, I., Kjartansson, O., Rivera, C., and Túbọ̀sún, K. Developing an Open-Source Corpus of Yoruba Speech. In *Proceedings of Interspeech 2020*, pages 404–408, Shanghai, China, October 2020. International Speech and Communication Association (ISCA).

[7] Heo, D., Rim, D. N., and Choi, H. N-gram prediction and word difference representations for language modeling. *arXiv preprint arXiv:2409.03295*, 2024.

[8] Jimoh, T. A., De Wille, T., and Nikolov, N. S. Bridging gaps in natural language processing for Yorubá: A systematic review of a decade of progress and prospects. *Natural Language Processing Journal*, page 100194, 2025.

[9] Kik, A., Adamec, M., Aikhenvald, A. Y., Bajzekova, J., Baro, N., Bowern, C., et al. Language and ethnobiological skills decline precipitously in papua new guinea, the world's most linguistically diverse nation. *Proceedings of the National Academy of Sciences*, 118(22):e2100096118, 2021.

[10] Mienye, I. D., Swart, T. G., and Obaido, G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9):517, 2024.

[11] Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[12] Noh, S. H. Analysis of gradient vanishing of rnns and performance comparison. *Information*, 12(11):442, 2021.

[13] Olorunfemi, T. O., Azubuike, O. L., and Ojo, O. E. On-screen keyboard with dictionary mapping for Yorùbá language. *The Journal of Computer Science and Its Applications*, 27(1):104–115, 2020.

[14] Oluokun, S. O., Ayobami, A. J., and Adefunso, A. Enhancing Yoruba text autocompletion with an attention-augmented recurrent neural network. *International Journal of Research and Innovation in Applied Science*, 10(10):1734–1752, 2025.

[15] Oyekanmi, E. O., Al-Turjman, F., and Fadare, O. Toward the realization of a high performing continuous yorùbá speech to text translation using a mobile phone. In *Artificial Intelligence Learning Facilitators*, pages 153–191. Auerbach Publications, 2025.

[16] Raeini, M. G. The evolution of language models: From n-grams to llms, and beyond. *Natural Language Processing Journal*, 12:100168, 2025.

[17] Singh, G. and Kamboj, C. P. Deep learning for predicting the next word in bilingual social media texts. *SN Computer Science*, 6(1):45, 2025.