Diacritic Restoration for Yoruba Text with under dot and Diacritic Marks Based on LSTM

*[1]Kingsley L.Ogheneruemu, [2]Jumoke F. Ajao, [2]Abdulrafiu M. Isiaka,
[3]Franklin O.Asahiah and [1]Olumide K. Orimogunje
[1]Department of Computer Science, Federal Polytechnic Ede, Nigeria
[2]Department of Computer Science, Kwara State University, Malete, Nigeria
[3]Obafemi Awolowo University, Ile-Ife, Nigeria.
aerokinghighest1@gmail.com | jumoke.ajao@kwasu.edu.ng | abdulrafiu.isiaka@kwasu.edu.ng

Received: 22-MAR-2023; Reviewed: 16-JUN-2023; Accepted: 18-SEPT-2023
http://doi.org/10.46792/fuoyejet.v8i3.1020

ORIGINAL RESEARCH

Abstract- Yoruba is a tonal language spoken primarily in Nigeria, some West African countries, and other parts of the world by over 40 million people. Many Yoruba texts written online lack tone marks, which can be confusing, ambiguous, and difficult for Natural Language Processing. This paper presents a method, which combines syllable-based approach and long short-term memory (LSTM) for diacritics restoration of standard Yoruba text. By enhancing the built-in varnishing gradient of RNN, the aim is intended to recover lost diacritics in Yoruba text for both characters that carry diacritic signs and underdot and return it with the proper diacritics. Data were acquired from Yoglobavoice, BBC Yoruba new and Yoruba words collected from literate indigenous writers. 27050 Yoglobalvoice datasets, 2000 Yoruba words extracted from BBC Yoruba news, and 1470 Yoruba words collected from a Yoruba language teacher. In addition, syllabic module was developed to group the tokenized word into different syllables. The output of the syllabication algorithm was fed into the Long Short-Term Memory (LSTM) module for training, the LSTM model was trained using 70% of the dataset and validated using 30% of the dataset. The result obtained showed 96% accuracy. From the result, it was observed that the use of LSTM for restoring diacritic gave an improved restoration of both character with under dot and character that contains tone-marks.

Keywords- Deep learning, Diacritic restoration, Image processing, Yoruba language

1 INTRODUCTION

Diacritic restoration is a technique for recovering diacritical marks in a written text (Dang & Nguyen, 2020). Writing without diacritics makes phrases harder to read; nonetheless, people do not write words with diacritics for a variety of reasons, including speed, convenience, or texting on devices that do not support diacritics (Nuţu *et al*, 2019; Náplava *et al*, 2018; Hifny, 2021). As a result, Natural Language Processing (NLP) tasks such as machine translation, sentiment analysis, and question answering systems struggle to process these messages. Hence, diacritics restoration is essential for future use or NLP activities (Asahiah, 2017).

The usage of diacritic-missing texts, which are typically ignored in many NLP applications, is made possible via diacritic restoration, diacritical languages; among these algorithms are naïve bayes, Recurrent Neural Network and Bayesian classifier. Despite these efforts, not much improvement on the performance of the diacritic restoration has been achieved (Ezeani, 2018). This research employed the use of deep learning approach to restore diacritic marks on Yoruba word, phrases and sentences. Also, the need to integrate other languages beside the English language into Information Technology to improve human-computer interaction necessitate the restoration of diacritics of tonal languages such as the Yoruba language.

Automating the process using computational model will facilitate efficient and accurate restoration. Existing studies on the restoration of Yoruba diacritics have extensively used machine learning approaches. This study explores the use of deep learning models given that they are more computationally powerful than conventional machine learning models. Also, existing studies on the Yoruba language partially restored only the tonal mark.

Deep Learning is a subfield of machine learning that uses an algorithm that draws inspiration from the structure of the human brain (Mariel *et al*., 2018). A neural network takes in information, trains itself to spot patterns, and then forecasts the results for fresh input. Layers of neurons make up neural networks, and these neurons serve as the network's central processing node (Náplava *et al*., 2019). Deep Learning consists of input layer which receives the input; the output layer predicts the outcome and in between is the hidden layer which performs most of the processing. The combination of multiple layers is what makes a neural network deep learning. Examples of deep learning algorithms include but not limited to, convolution neural networks (CNN), recurrent neural networks (RNN), Long Short-Term Memory networks (LSTMS), deep belief networks (DBN), Generative adversarial networks (GAN) (Hifny, 2018).

Yoruba is a tonal language spoken by some West African countries and other parts of the world, with a population of over 40 million people (Asahiah *et al*., 2017, Ajao *et al*., 2018). Many Yoruba texts written online lack tone marks, resulting in poor performance in the interpretation of some of the words (Asahiah et al., 2017; Hucko, & Lacko, 2018). The need to overcome the syntactic and semantic problems caused by a lack of or incorrect usage of diacritics, especially in low-resource languages, prompted the

*Corresponding Author

Section B- ELECTRICAL/COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Ogheneruemu K. L., Ajao J. F., Isiaka A. M., Asahiah F. O. and Orimogunje O. K., (2023). Diacritic Restoration for Yoruba Text with under dot and Diacritic Mark Based on LSTM, FUOYE Journal of Engineering and Technology (FUOYEJET), 8(3), 284-293. http://doi.org/10.46792/fuoyejet.v8i3.1020

majority of diacritic restoration efforts (Ezeani, 2019). Thus, this study is designed to restore missing diacritics in Yoruba text and return it with correct diacritics. The proposed system used combined syllable-based approach and long short-term memory (LSTM) for diacritics restoration of standard Yoruba text. This research work is intended to contribute to the body of knowledge in the following:

- Addition of under-dot tone mark to existing diacritics system.
- An LSTM deep learning model using syllable-based approach to improve on the varnishing gradient of RNN.

Recently, several approaches have been used to restore diacritic on text. (Ezeani *et al.*, 2018) proposed Igbo diacritic restoration using embedding models. (Orife, 2018) proposed Attentive sequence-to-sequence learning for diacritic restoration of Yoruba Language text based on recurrent neural network. The usage of LSTM was a solution to the recurrent neural network's inherent varnishing gradient problem. In this piece, characters with tone marks and underdots are restored utilizing long short-term memory networks (LSTMS).

2 Theoretical Background

2.1 Corpus Construction From Legacy Text

Existing texts in several diacritic languages were usually saved in non-standard forms. Since the absence of diacritics rarely makes a document impossible for a human reader to read, this lack of text standards has persisted and been tolerated. Due to a high amount of ambiguity induced by the lack of diacritics, such writings are typically easy to read by humans but difficult to comprehend using language tools. Despite the fact that these texts are extremely valuable to native speakers, their poor quality prevents them from being useful in improving NLP research and development. Diacritic restoration can be used to recover and standardize legacy texts in order to create useable corpora (Ezeani, 2019).

2.2 Deep Learning Algorithms

The structure of the human brain serves as the basis for the artificial intelligence subset known as neural networks. It is made up of layers of neurons. The first layer is the input layer which receives an input and the last layer which is referred to as output layer gives the final prediction. Between the input and the output layer is the hidden layer which perform most of the processing. A neural network having multiple of this layer is called deep learning. In the next section the types of deep learning algorithm will be discussed (Hung, 2018).

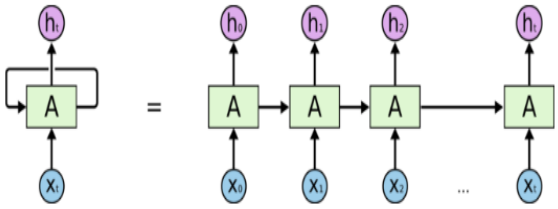
2.2.1 Convolutional Neural Network (CNN)

Convolutional neural network is special kind of neural network popularly used for analysing, classifying images and natural language processing (NLP). CNN is specialized in picking out or detecting patterns in data. A CNN has hidden layers called convolutional layers in which the number of filters can be specified. Filters are the component responsible for picking out patterns. A convolutional layer operation involves placing a filter over an array of pixel. In this process the convolved feature map is created. The pooling layer helps speed up processing by reducing the number of parameters the network needs, which in turn reduces the sample size for a particular feature map. This produces a pooled feature map. The maximum input from the convolved features is used in max pooling, and the average input is used in average pooling (Hung, 2018, Hifny, 2018).

2.2.2 Recurrent Neural Network (RNN)

Given that RNN contains an internal memory that enables it to recall its input; RNN is well suited for machine learning tasks that call for sequential data. All of the inputs in an RNN are interconnected, as seen in Figure 2.1. Given an input sequence of (X 0...X n), the RNN takes in X(0) and outputs h(0), which, along with X(1), serves as the input for the following phase. Therefore, the input for the following step is h(0) and X(1). The input for the subsequent stage is h(1) and X(2), and so forth.

Fig. 1: Recurrent Neural Network (Aditi, 2019)



The current state can be represented in the following equation (Aditi, 2019):

$$h_t = f(h_{t-1}, X_t) \quad (1)$$

Adding the activation function, the equation becomes (Aditi, 2019) :

$$h_t = tanh(W_{hh}h_{t-1} + W_{Xh}X_t) \quad (2)$$

Where h denotes the single hidden vector, W represents the weight, ,W_{hh} refers to the weight at previous hidden state, and W_{Xh} is the weight at current input state, tanh is the Activation funtion,

$$Y_t = W_{hY}h_t \quad (3)$$

Y_t is the output state. W_{hY} represent the weight at the output state.

2.2.3 Long Short-Term Memory (LSTM)

RNN a deep learning algorithm that lacks long-term memory and that is the reason for the development of LSTM. The LSTM is a particular kind of RNN that is appropriate for learning from significant events with very long lags. LSTM is made up of a cell, an input gate, an output gate, and a forget gate. The three gates control the flow of information in and out of the cell, and the cell remembers values across arbitrary time interval. The cell remembers values across arbitrary time intervals, thanks to the three gates that control the flow of information in and out of it. The mathematical model for LSTM is given in equations 4 to 8.

$$i_k = \sigma(W_1 h_{k-1} + U_1 X_k) \quad (4)$$
$$f_k = \sigma(W_f h_{k-1} + U_f X_k) \quad (5)$$
$$O_k = \sigma(W_o h_{k-1} + U_o X_k) \quad (6)$$

http://doi.org/10.46792/fuoyejet.v8i3.1020 journal.engineering.fuoye.edu.ng/

$$C_k = f_{k\odot}c_{k-1} + k \odot \tanh(Wch_{k-1} + U_cX_k) \quad (7)$$
$$h_k = O_{k\odot}\tanh(C_k) \quad (8)$$

where i_k denotes the input gate and f_{tk} denotes the forget gate The output gate is O_k. The memory cell is C_k, and the hidden state is h_t. $\odot$ signify element-by-element multiplication. W and U denote weight matrices and bias vector parameters, respectively, that must be learned during training.

2.3 Difference between RNN and LSTM

RNN is capable of accepting input in the form of sequence. However, training RNN on problems requiring long-term temporal relationships is a difficult task. This is due to the fact that the gradient of the loss function decays exponentially over time (called the vanishing gradient problem). In LSTM a short-term memory is added which makes it easier to remember past data. The exploding problem is fixed by decoupling cell state c and hidden layer/output h, which makes memories in c more stable. The disappearing gradient is resolved using an improved form of backward propagation called "constant error back propagation." As a result, it is difficult for the gradient flow via c to disappear (therefore the overall gradient is hard to vanish). LSTM is characterized by three gates namely Input, forget, and output gate (Aditi, 2019; Hifny, 2018.).

2.4 Related Works

Alqahtani *et al*., (2020) proposed using convolutional neural networks to restore the diacritical marks. The results of the study demonstrated that character-based convolutional architectures for diacritization produce comparable results to both word- and character-based RNN ones for a variety of languages, including Arabic, Yoruba, and Vietnamese, though at a substantially reduced computational cost. Moreover, character-based modelling yields better performance overall for the diacritization task. Using future information is essential in diacritization since A-TCN consistently outperforms TCN, with a reduction in error rate of up to 40%. It was discovered that A-TCN offers accurate, effective answers.

Laki & Yang, (2020) proposed automatic diacritic restoration. machine translation for east-central European languages using a transformer model in the study, they developed a method for restoring diacritical marks based on cutting-edge neural machine translation (NMT) approaches (transformer model and sentence piece tokenization) 14 languages from the East-Central European region were used to test the system. The majority of system performances are greater than 98%.
Masmoudi *et al*., (2019) presented an automatic diacritic restoration for the Tunisian dialect. A statistical machine translation (SMT) and a discriminative model for a sequence classification task based on Conditional Random Fields (CRF) were the two main models created. In the second method, POS features were added to affect the diacritics generation. Both the word and character levels, restorations were carried out. Word error rates (WER) for CRF and SMT were both quite high (21.44% and 34.6%, respectively), demonstrating the effectiveness of automatic diacritization based on the CRF approach.

Hifny, (2019) introduced a byte pair encoding (BPE)-based open vocabulary tokenization method for the restoration of Arabic diacritics. The BPE approach divides the words into sub-word units of varying length and permits open vocabulary from the dictionaries of fixed sub-word units. The Tashkeela diacritization task findings demonstrate that the suggested method performs better than the character-based approaches. Orife, (2018). Based on careful sequence-to-sequence learning, corrected diacritical marks were added to Yoruba language text. Two separate attentive Sequence-to-Sequence neural models were used in the paper's studies to process undacritized material. The accuracy score—which was calculated as the proportion of correctly restored words to all words—was used to assess how well the generated models performed. Based on the test set targets, the perplexity of each model's predictions was determined. The method yields diacritization error rates of less than 5% on the evaluation dataset.

Ezeani *et al*., (2018) presented Igbo diacritic restoration using embedding models. 29 wordkeys with various variations were utilized. A list of sentences with blanks to be filled in with the appropriate wordkey variant was kept for each wordkey, excluding punctuation and digits. The word embedding models utilized fell into two categories: those trained using the Igbo bible corpus and those projected from the English embedding space. In comparison to the baseline n-gram models, the experimental results reveal an accuracy of 82.49%. Hifny, (2018). Developed a hybrid BiLSTM/MaxEnt tagger for syntactic diacritic restoration. For a group of Arabic words, the tagger assigns the syntactic diacritics. The work takes into account the fact that the syntactic diacritical marks serve only as a function of the words and their Arabic grammar. This hybrid LSTM/MaxEnt strategy outperforms the n-gram baseline models on the Arabic tree bank test. Asahiah et al. (2017). Developed tone-mark restoration for Yoruba text that was compiled from several sources was handled. A total of 250,336 words were analysed. Syllabication of these words produced 464,274 syllables. Various ratios of the syllables were employed for training and testing, ranging from 99% for training and 1% for testing to 70% for training and 30% for testing. In a ten-fold cross validation test, it was shown that using 75% of the data for training and the remaining 25% for testing produced the least variable results. Syllables are used as the processing linguistic units in this method, which can be combined with other techniques like lexicon lookup to get results that are likely to be better than the existing one.

Oladipo, (2017) formulated, implemented and evaluated a computational system for restoring missing diacritics for the Standard Yoruba digital text. This was with a view to automate the processing that renders Yoruba in standard orthography. A computational model was formulated as similarity and probabilistic based learning using supervised learning tools. software for diacritic restoration was designed using Unified Modeling Language and implemented. The system was evaluated using Grapheme Error Rate (GER), Syllable Error Rate (SER) and Word Error Rate (WER) for dot-below, tone-marks and word-level restorations respectively. The study showed that diacritic restoration in standard Yoruba text was better

http://doi.org/10.46792/fuoyejet.v8i3.1020 journal.engineering.fuoye.edu.ng/

accomplished by sequential restoring dot-below before tone marks. Šanti & Šnajder, (2009)employed a word-based method diacritics restoration in Croatian text were recovered. The system is cheap to compute, does not require any pre-processing, and relies on a dictionary and a bigram language model. Evaluation reveals that while adding a language model raises restoration accuracy to approximately 99%. Rotimi, (2005) developed a diacritic recovering system for tone mark restoration in Yoruba text. The paper provided a general background to the issue by emphasizing its significance and scope in Yorùbá, the case study language. A review of the current strategies was done after that. The proposed data-driven, syllable-based approach for tone marks restoration in Yorùbá text was then presented. The key stages in the development of the proposed model are the online training of the tone marks model from data using supervised learning. The second stage is the tone mark restoration system where the process starts with text to be tone marked are passed to the syllabification module. In adopting an approach to diacritic restoration.

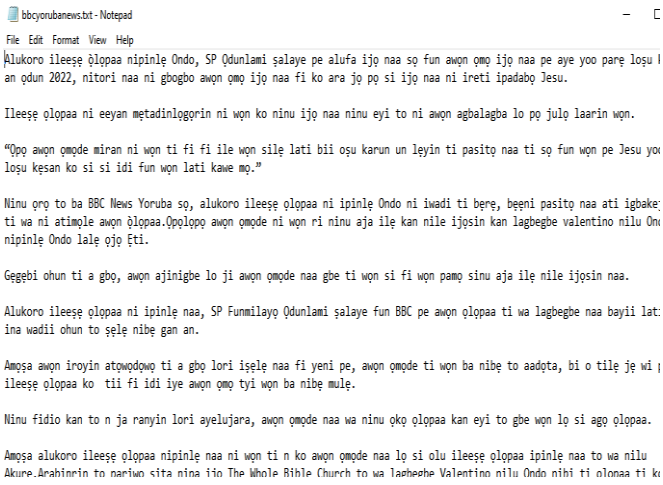
2.5 Inference Deduced From Literature

According to the reviewed literature, only (Asahiah *et al*., 2017) and (Orife, 2018) focused on restoring diacritics in Yoruba language. The two authors addressed tone-mark restoration as a subset of diacritic restoration but did not consider under-dot restoration. Therefore, this research presents a deep learning algorithm (LSTM) for diacritic restoration of under-dot and diacritic mark in Yoruba text.

3 Methodology

3.1 Data Acquisition, Aggregation And Normalization

Data was acquired from three different sources, Yoglobalvoice datasets, Yoruba words extracted from BBC Yoruba news and Yoruba words collected from a Yoruba language teacher was used for training and test in this research. Yoglobalvoices is a corpus of journalistic news text from a multilingual community journalists, translators, blogger, academics and human right activist. The datasets acquired from the three sources were aggregated into a text file as one. Also, the aggregated data was taken through normalization process which includes removal of duplicate words and transforming all the text to lower case. This was achieved through NLTK python NLP library. Before performing diacritics restoration in the LSTM model, the datasets was first passed through series of preprocessing stages which involves tokenization and syllabification. Finally, the model was trained, tested and evaluated in terms of accuracy.


bbcyorubanews.txt - Notepad

File Edit Format View Help

Alukoro ileeṣẹ ọlọpaa nipinlẹ Ondo, SP Ọdunlami ṣalaye pe alufa ijọ naa sọ fun awọn ọmọ ijọ naa pe aye yoo parẹ losu
an ọdun 2022, nitori naa ni gbogbo awọn ọmọ ijọ naa fi ko ara jọ pọ si ijọ naa ni ireti ipadabọ Jesu.

Ileeṣẹ ọlọpaa ni eeyan mẹtadinlogorin ni wọn ko ninu ijọ naa ninu eyi to ni awọn agbalagba lo pọ julọ laarin wọn.

"Ọpọ awọn ọmọde miran ni wọn ti fi fi ile wọn silẹ lati bii oṣu karun un leyin ti pasitọ naa ti sọ fun wọn pe Jesu yo
losu kẹsan ko si si idi fun wọn lati kawe mọ."

Ninu ọrọ to ba BBC News Yoruba sọ, alukoro ileeṣẹ ọlọpaa ni ipinlẹ Ondo ni iwadi ti bẹrẹ, bẹẹni pasitọ naa ati igbake
ti wa ni atimọle awọn ọlọpaa.Ọpọlọpọ awọn ọmọde ni wọn ri ninu aja ilẹ kan nile ijọsin kan lagbegbe valentino nilu On
nipinlẹ Ondo lalẹ ọjọ ẹti.

Gẹgẹbi ohun ti a gbọ, awọn ajinigbe lo ji awọn ọmọde naa gbe ti wọn si fi wọn pamọ sinu aja ilẹ nile ijọsin naa.

Alukoro ileeṣẹ ọlọpaa ni ipinlẹ naa, SP Funmilayọ Ọdunlami ṣalaye fun BBC pe awọn ọlọpaa ti wa lagbegbe naa bayii lat
ina wadii ohun to ṣẹlẹ nibẹ gan an.

Amọṣa awọn iroyin atọwọdọwọ ti a gbọ lori isẹlẹ naa fi yeni pe, awọn ọmọde ti wọn ba nibẹ to aadọta, bi o tilẹ jẹ wi
ileeṣẹ ọlọpaa ko tii fi idi iye awọn ọmọ tyi wọn ba nibẹ mulẹ.

Ninu fidio kan to n ja ranyin lori ayelujara, awọn ọmọde naa wa ninu ọkọ ọlọpaa kan eyi to gbe wọn lọ si agọ ọlọpaa.

Amọṣa alukoro ileeṣẹ ọlọpaa nipinlẹ naa ni wọn ti n ko awọn ọmọde naa lọ si olu ileeṣẹ ọlọpaa ipinlẹ naa to wa nilu
Akure.Arabinrin to nagiwo sita ninu iio The Whole Bible Church to wa lagbegbe Valentino nilu Ondo nibi ti olopaa ti k

Fig. 2: Research Dataset

3.2 The Formulated Model

The proposed model employs various modules which include tokenization, syllabification, and convergence module. Firstly, the dataset is passed through the pre-processing stage which involves tokenization and syllabification after which the result of preprocessing was passed to the LSTM model. The proposed model is as illustrated in Figure 3.

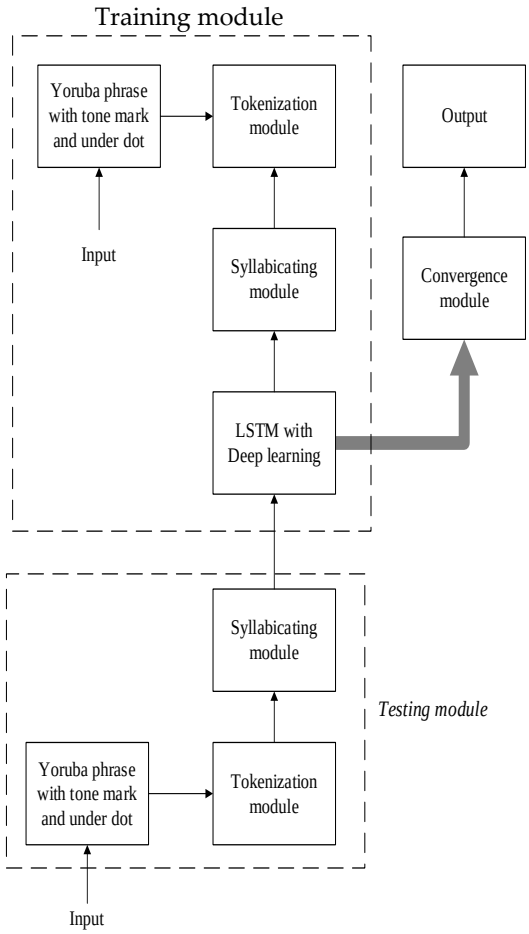


Fig. 3: Sequence to Sequence Stacked LSTM Model Architecture

3.3 Tokenization Module

Tokenization is a process of splitting the data to form a huge vector of word. This involves splitting the document by white space, new lines, tabs and saving it again. This

was implemented in python by using genism natural language processing framework.

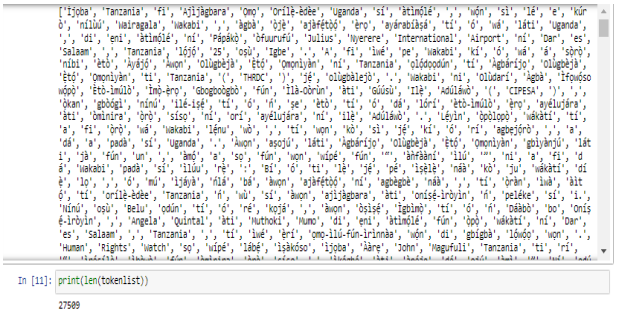


Fig 4: Tokenization of Training Dataset

3.4 SYLLABICATING MODULE

This involve breaking each word into its constituent syllables. The syllabification module adopted in this study is illustrated in the Algorithm 1. The module operates by iterating through the input Yoruba word to check for possible combination of characters that could result to syllable. Syllables are formed considering the following four circumstances;

i) When a vowel precedes a consonant (starts a word) the first vowel letter will be taken as a syllable.
ii) When a consonant precedes a vowel in a word, the consonant together with the vowel is taken as a syllable.
iii) When two consonant precedes a vowel, the consonants together with the vowel is taken as a syllable.
iv) When two consecutive characters are vowel (disyllable) precedes a consonant, the vowels are taken as a syllable.

Algorithm 1. Syllabification Module

Input word W
Output syllableList[ ]
Step 1: compute length L of the word len(word)
i=0
Step 2: While(i< L) {
i++
Step 2.1: if W[i] &&W[+1]== vowel{
syllableList.append(W[i]+W[i+1])
}
Step 2.2: if W[i]==vowel && W[i+1]== consonant{
syllableList.append(W[index])
}
Step 2.3: if W[i]==consonant && W[i+1]== vowel{
syllableList.append(W[i]+W[i+1])
}
Step 2.4: if W[i]==consonant&& W[i+1] == consonant && W[i+2]==vowel{
syllableList.append(W[i]+W[i+1]+W[i+2])
}
output syllableList

According to Algorithm 1 the syllabification module starts by accepting Yoruba word as input. The algorithm then computes the word length and loop through each letter of the word to examine possible combination of Yoruba syllable using if statement. There are four if statement in the algorithm which determine different combination of letter that could result to a syllable and then append the syllable in a syllable list.

3.6 LONG SHORT-TERM MEMORY (LSTM)

LSTMs were created to solve the problem of gradient vanishing inherent in RNN. Gated Recurrent Units (GRUs), like LSTMs, were created to deal with long-term dependencies. A memory cell and three gating units make up an LSTM network. In Figure 5, the input gate controls how much information is added to the current memory, the forget gate controls how much information is removed from the previous memory, and the output gate controls how much information is output from the current memory. These gates accept the previous concealed state and the current input as inputs and produce a number between 0 and 1 as an output (i.e., logistic function). The multiplication of the output of these gates controls the flow of information into and out of the memory.

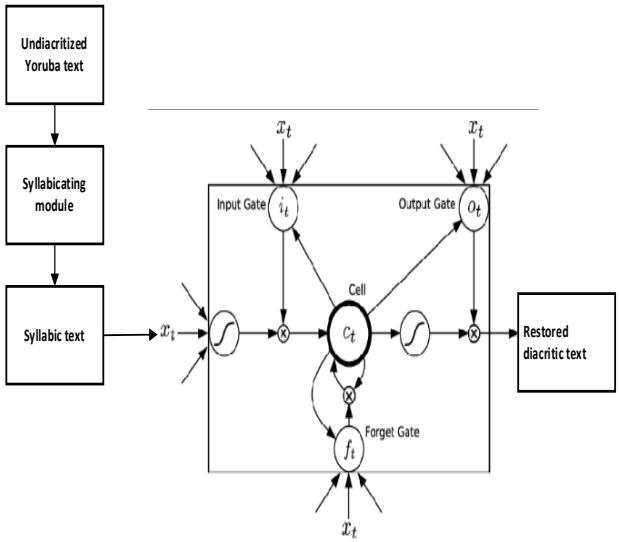


Fig. 5: LSTM Model

3.6.1 Learn Gate

Event and previous Short-Term Memory were used as input, and only pertinent data is retained for prediction. The current event vector and the weight matrix from the prior short-term memory are multiplied together in order to introduce non-linearity, with the bias provided by the tanh (hyperbolic Tangent) function. The matrix is produced as a result. A biased sigmoid activation function is used to pass the result of merging the short-term memory and current event vectors, multiplying it by a weight matrix, and then using the result to disregard unnecessary information. The learn matrix is created by multiplying the ignore factor by the learn gate result.

3.6.2 Forget Gate

The Previous Long-Term Memory is used by the Forget Gate to determine which knowledge should be kept and which should be lost. When the current event vector Et and prior short-term memory are merged, multiplied by the weight matrix, and then run through the biased Sigmoid activation function. By dividing the forget factor by the prior long-term memory, the forget gate's output is determined.

3.6.3 Remember Gate

Recall Gate combines the current experience with past short-term memory to produce output. The outputs of the Forget Gate and Learn Gate are combined to produce the Remember Gate's output, which is LTM for the cell after it.

3.7 Implementation of the LSTM Model

The LSTM model was implemented in python programming language, keras framework and tensor flow as the backend. The learning rate was set to 0.01, batch size 128, number of epochs 100, dropout rate 0.2. The model is a 6 layered LSTM with 1.7 million parameters and number of neurons is 256. The LSTM takes in input from the sequencing stage of the preprocessing. For clarity, the model's hyper-parameter is represented in the table shown Table 1.

Table 1. Hyperparameters of LSTM Model

Hyperparameter	Values
Learning rate	0.01
Batch size	128
No of epochs	100
Dropout Rate	0.2p
Activation function	RELU
Activation function output layer	Softmax
Optimizer	Adam

3.8 LSTM System Performance

The model was evaluated based on word level accuracy and word error rate(WER). The accuracy of a prediction is defined as follows:

$$Accuracy = \frac{Number\ of\ correct\ words}{Number\ of\ words} \quad (1)$$

$$WER = \frac{Number\ of\ incorrect\ words}{Number\ of\ words} \quad (2)$$

4 Result and Analysis

The results presented include data preprocessing result, evaluation of the developed model, and comparison with existing future work. Niger-volta dataset was used for analysis. The datasets were passed through the tokenization and word embedding and before passing into the LSTM model. The result of the preprocessing stage is a digital vector that was passed afterwards to the LSTM model. The Figure 6 present the first layer type of the embedding layer with 126300 parameters. The embedding layer convert each syllable to a fixed length of vector. Apart from the embedding layer there are other two layers namely LSTM and dense layer which have 3488 and 9 parameters respectively. The total number of parameters is 129,799.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 313, 100)          126300
_________________________________________________________________
lstm_3 (LSTM)                (None, 8)                 3488
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 9
=================================================================
Total params: 129,797
Trainable params: 129,797
Non-trainable params: 0
_________________________________________________________________
None
<keras.engine.sequential.Sequential at 0x7f2978304dd0>
```

Fig. 6. Embedding Layer

4.1 Result of Tokenization Module

The result of tokenization module is presented in Figure 7. The dataset contains Yoruba phrases which were broken down into words or token. After the tokenization process the number of tokens in the dataset was found to be 27,509.

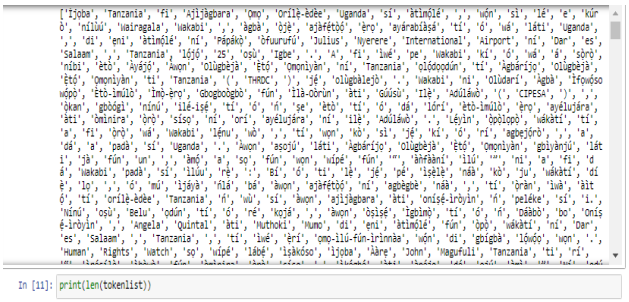


Fig. 7: Tokenization Module Analysis

4.2 Result of Syllabicating Module

The result of syllabicating module is as shown in Figure 8. In the syllabicating module, each word was split into their constituent syllables. This was computed based on the algorithm in algorithm 1.

```
['`i', 'jọ', 'ba']
['a', 'jì', 'jà', 'gba', 'ra']
['Ọ', 'mọ']
['o', 'rí', 'lẹ̀', 'è', 'dèe']
['à', 'tì', 'mọ́', 'lé']
```

Fig. 8. Syllabicating Module Analysis

4.3 Test Cases for Dataset

The developed model was subjected to test. The process of testing involves inputting undiacritized Yoruba text from a text file into the model and the expected output is a Yoruba text which was diacritized. The process is as shown in Figure 9.

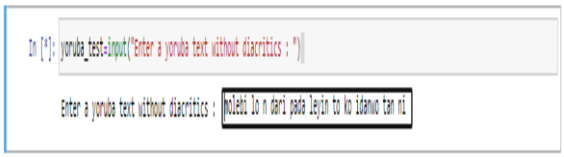


Fig. 9: Undiacritized Yoruba Text

The input is taking through preprocessing stage before being input into the LSTM model. The expected output of this model is a diacritized Yoruba text. The output is as shown in Figure 10.

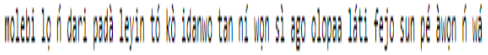


Fig. 10: Diacritized Yoruba Text

Apart from sentences, the model was also tested with Yoruba phrases and words. The test on phrases is shown in Figure 11. The Yoruba text only contains two words which are not diacritized.

Fig. 11: Test at Phrase Level

The output is the diacritized Yoruba phrase which is as shown in Figure 12.

Fig. 12: Result of Test at Phrase Level

The model was tested with Yoruba words that have variants; an example is 'igba' which can assume different meaning based on the diacritics used. The test on Yoruba word is shown in Figure 13.

Fig. 13: Test at Word Level

The output of this test is variants of the word with their respective diacritics. The output is shown in Figure 14.

Fig. 14: Result of Test at Word Level

4.3.1 Dataset 1 (Yoruba Teacher)

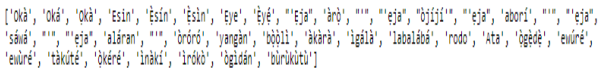


Fig. 15: Training Dataset

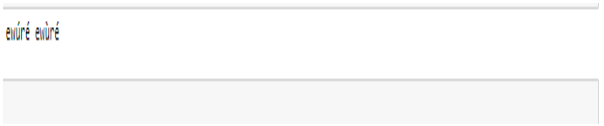


Fig. 16: Result of Test Dataset

4.3.2 Dataset 2 (BBC Yoruba news)

This contains 926 tokens or words with test results shown in Figure 17.

Fig. 17: Result of BBC Test Dataset

Finally, the model was tested with Yoruba words, phrases and sentences that have under-dot. The test on Yoruba word is shown in Figure 18.

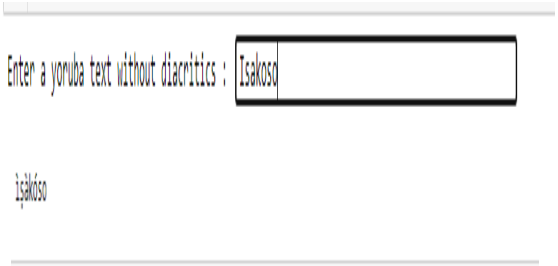


Fig 18: Result of Test of Words with Under-Dot

The test of Yoruba phrases with under-dot was also carried out as shown in Figure 19.

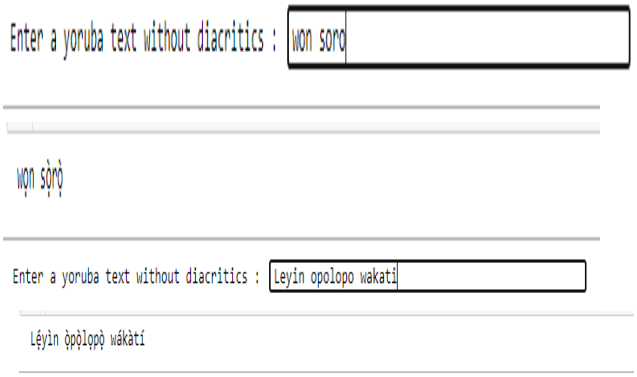


Fig 19: Result of Test Phrase with Under-Dot

The test of Yoruba sentences with under-dot is shown in Figure 20.

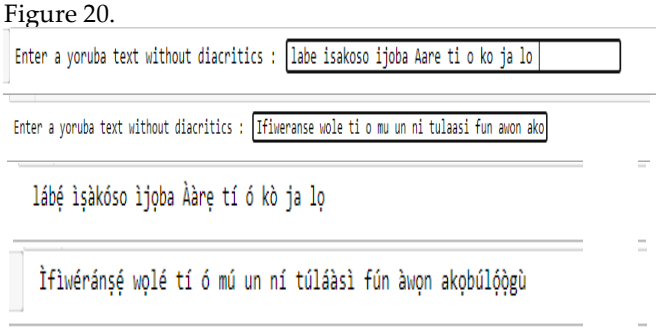


Fig. 20: Results of Test Sentences with Under-Dot

5 Discussion

In other to evaluate the performance of the developed system, the dataset was divided into two partitions namely training set (70%) and testing set (30%). The total number of sentences in the dataset used is 910. The system was tested with word that has different variants and also with 2-gram, 3-gram and n-gram. The result is as shown in Tables 3 and 4 respectively. The system was evaluated based on word level accuracy. Firstly, the test was carried out on Yoruba words that have different variants. The aim here is to determine the number of variants whose diacritics are restored correctly and the number of variants whose diacritic are correctly restored. Result of the test is presented in the Table 2.

Table 2. One Word Analysis

Word	Variant	Restored	Not Restored
Oko	6	6	0
Igba	4	4	0
Owo	3	3	0
Iba	2	1	1
Ara	3	3	0
Emi	2	2	0

http://doi.org/10.46792/fuoyejet.v8i3.1020 journal.engineering.fuoye.edu.ng/

Secondly, the developed model was tested on one gram. The test result is presented in Table 6.2. Out of 10 words that were tested, diacritics of 9 words were restored correctly while 1 was not restored.

Table 3. One gram Analysis

Word	Diacritized word	Number of words	Restored	Not Restored
Opolopo	ọ̀pọ̀lọpọ̀	1	1	0
Owo	owó	1	1	0
Ijaya	ìjáyà	1	1	0
Ibowo	ìbọ̀wọ̀	1	1	0
Aare	àárẹ	1	1	0
Adalu	àdàlù	1	1	0
Agbegbe	agbegbe	1	1	0
Iberu	ìbẹ̀rù	1	1	0
itesiwaju	ìtẹ̀síwájú	1	1	0
Awon	àwọn	1	1	0

The third test involve 2-gram Yoruba word, among the ten 2-gram words that were tested, one word was correctly restored in 7 cases while 2 words were correctly restored in 3 cases. Diacritics was not restored in 3 cases.

Table 4. Two-gram Analysis

Word	Diacritized word	Number of words	Restored	Not Restored
Omo mi	ọmọ mi	2	1	1
Ile mi	ilé mi	2	1	1
Ile eko	iléeko	2	1	1
Iselenla	ìṣẹ̀lẹ̀ ńlá	2	2	1
Erooja	ẹ̀rọ̀oja	2	1	1
itesiwajuwa	ìtẹ̀síwájúwá	2	2	0
Omo ale	ọmọ ale	2	1	1
Omoorile-edee	Omoorílẹ̀-èdèe	2	1	1
Awononise-iroyin erookan	Awononíṣẹ́-ìròyìn ẹ̀rọ̀ọkàn	2	1	1
		2	2	1

In the fourth test, 3-gram Yoruba words were tested and the result is presented in the Table 6.4. The model was able to restore diacritics for two words in 6 cases while 3 3-gram were completely restored. The total number of words restored correctly is 13.

Table 5. Three grams Analysis

Word	Diacritized word	Number of words	Restored	Not Restored
Okoojuomi	Okoojúomi	3	1	2
Bo bata re	Bo bataré	3	1	2
Gbe mi soke	Gbé mi soke	3	1	2
Fi ka le	Fi ka lẹ̀	3	3	0
Awonosiseigbimo	àwọnòṣìṣẹ́ igbimo	3	3	0
Wo aso re	Wo asoré	3	2	1
Bamigbesibi	Bamigbésibi	3	1	2
Emi lo kan	Emi lọkan	3	1	2
Mo n bo	Mo ń bo	3	1	2
Rin wasibi	Rin wásibi	3	3	0

The model was also tested on n-gram. The result is presented in Table 6.5. The total number of words involved in the test is 85. The total number of words whose diacritics mark were restored is 78 while 7 words were not restored.

Table 6. n-gram Analysis

Word	Diacritized word	Num	Restored	Not Restored
Bi o ti le je pe	Bí ó tilẹ̀ jẹ́ pé	6	6	0
Iselenaa ko juwakati di e lo	ìṣẹ̀lẹ̀ náàkòjuwákàtídíẹ̀ lọ	8	8	0
O mu Ijaya n la baa won a jafetooni a gbegbenaa	ó múìjáyàńlábáàwọnajàfẹ́tọ̀ọ́ níagbègbènáà	13	10	3
Awonajijagbaraationise-iroyin n pelekesi	àwọnajìjàgbaraàtionísẹ́-ìròyìn ń pelékesí	7	7	0
Ninuodunti o re koja	Nínúọdúntí ó rékọjá	6	5	1
AwonosiseIgbimotio ndaaboboonise-iroyin	àwọnòṣìṣẹ́ Ìgbìmọ̀ tí ó ń DáàbòboOnísẹ́-ìròyìn	11	11	0
Ipalenumoileeseakoroyi n aladaaduronipaseawonadaluisiwee fun ileiseakoroyin	Ìpalẹ́numọ́ iléẹṣẹ́ akọ̀ròyìnaládàádúrónípaseẹàwọnàdàlùìṣíwèéfúniléiṣẹ́ akọ̀ròyìn	13	13	0
Atiirokekemoonise-iroyinti di ideruba	àtiìrókẹ́kẹ́ mọ́ onísẹ́-ìròyìnti di ìdẹ́rùbà	7	6	1
Ikora-eni-ni-ijanuatiIberu latiso eruokan	ìkóra-ẹni-ní-ìjánuàtiìbẹ̀rùlátisọ̀èròọkàn	7	7	0
Ikegbeati a peju de ojuami	ìkẹ́gbẹ́ àtiàpéjọdéojúàmì	7	5	2

Finally, the model was tested with Yoruba words that contain under-dot. The total number of words tested is 25, among which 24 were restored word were correctly restored. The result is shown in Table 6.6.

Table 7. Result of Test for Yoruba Text with Under-Dot

Yoruba Text	Diacritized word	Num word	Restored	Not Restored
Isakoso	ìṣàkóso	1	1	0
Won soro	wọ́nsọ̀rọ̀	2	2	0
Leyinopolopo wakati	Lẹ́yìnọ̀pọ̀lọpọ̀ wákàtí	3	3	0
Labe isakosoljoba Aare tiooko ja lo	lábẹ́ ìṣàkósoìjọbaÀàrẹtí o kò ja lọ	8	7	1
Ifiweransewoleti o mu un nitulaasi fun awonakobulo ogu	Ìfiwéránṣẹ́ wọ́létí ó mú un nítúláàsìfúnàwọnakọbúlọ́ọ̀gù	11	11	0

The result of the performance evaluation is presented in Table6.8. The model reported word level accuracy of 95.00% for test that involve one word and variants. One Gram, Two-Gram, Three Grams, and n-Grams were also tested with word level accuracy of 90.00%,65.00%,63.00% and 91.76% respectively. Finally, for test that include underdot, the model reported an accuracy of 96.00%.

Table 8. Performance Evaluation of the Developed Model

Test	Word Level Accuracy
One word	95.00%
One Gram	90.00%
Two Gram	65.00%
Three Gram	63.33%
N gram	91.76%
Under-dot	96.00%

For clarity, the performance evaluation is presented in the bar chart in Figure 21.

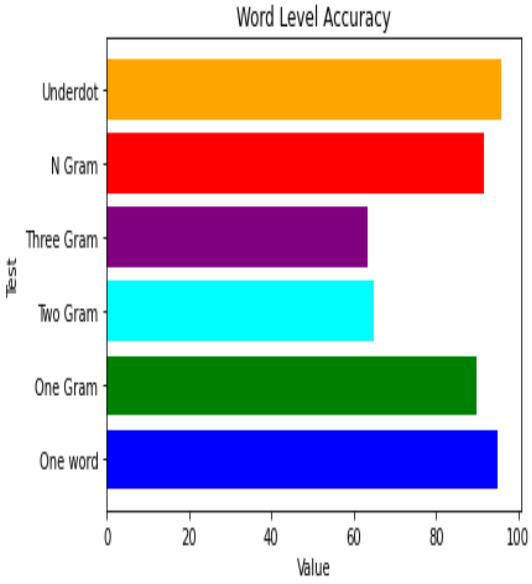


Fig. 21: Performance Evaluation Chart of the Developed Model

Apart from word level accuracy the model was also evaluated in terms of word error rate and the result is also presented in Figure 22.

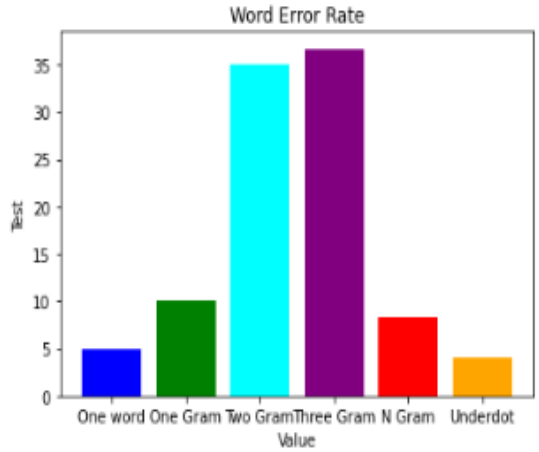


Fig. 22: Performance Evaluation Chart of the Developed Model using Word Error Rate

5.1 Comparison with the Existing Works

In order to further evaluate the model's performance, it was compared with existing works. Ezeani et al (2018) employ word embedding model for diacritics and achieved an accuracy of 82.49% .Also, the model was compared to Asahiah et al., (2017) in which syllable based approach was adopted yielding a result of 96.23% accuracy. Comparing Ezeani et al., (2018) result with the result achieved in this study shows a significant increase in accuracy; comparative result is presented in Table 9.

Table 9. Comparison with Related Work

Research	Methodology	Result (Accuracy)
Ezeani *et al.*, (2018)	Embedding model	82.49%
Naplava *et al.*, (2019)	Word based	76%
Hung, (2019)	Word based	95%
Laki& Yang, (2020)	Word based	99.8%
The Formulated model	Syllable based	96.00%

6 Conclusion and Future Work

In this research LSTM model was used to restore diacritics in Yoruba text. Dataset from Yoglobalvoice, BBC Yoruba News and words written by Yoruba Teacher was used for testing and training the developed LSTM model. The dataset was taken through preprocessing stage which involves tokenization, The developed LSTM System takes preprocessed Yoruba text data in as input for training. The System's performance was evaluated in terms of accuracy and Word Error Rate. Comparison with related work show that the result obtained in this study is far better. There are other deep learning models such as CNN, MLP. Future work should consider including these deep learning algorithms and performing comparative analysis. Based on the fact that most available texts for development lack a significant number of diacritics, the requirement to restore diacritics frequently arises in Natural Language Processing (NLP). Because diacritics can affect the meaning or pronunciation of particular words, the accuracy of grammar may be in question if they are not used correctly. It is also claimed that, in addition to degrading the language in and of itself, the absence of diacritics is a significant obstacle to language processing activities..Diacritics restoration has application in NLP. The findings of this research will be useful for expert working in this field to create NLP software. It will also enhance user's speed in typingYoruba sentences.

References

Aditi, M. (2019). *Understanding RNN and LSTM. What is Neural Network? | by Aditi Mittal | Medium*. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

Alqahtani, S. (2020). *Full and Partial Diacritic Restoration: Development and Impact on Downstream Applications*. *May 2009*.

Alqahtani, S., Aldarmaki, H., & Diab, M. (2019). Homograph disambiguation through selective diacritic restoration. In *ACL 2019 - 4th Arabic Natural Language Processing Workshop, WANLP 2019 - Proceedings of the Workshop* (pp. 49–59). https://doi.org/10.18653/v1/w19-4606

Alqahtani, S., Mishra, A., & Diab, M. (2020). Efficient convolutional neural networks for diacritic restoration. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 1442–1448. https://doi.org/10.18653/v1/d19-1151

Asahiah, F. O., Odéjobí, O. À., &Adagunodo, E. R. (2017). Restoring tone-marks in standard Yorùbá electronic text: Improved model. *Computer Science*, *18*(3), 301–315. https://doi.org/10.7494/csci.2017.18.3.2128

Dang, T. D. A., & Nguyen, T. T. T. (2020). TDP – A Hybrid Diacritic Restoration with Transformer Decoder. *Proceedings of the 34th*

Pacific Asia Conference on Language, Information and Computation, 76–83. https://aclanthology.org/2020.paclic-1.9

Ezeani, I. (2019). *Corpus-Based Approaches to Igbo Diacritic Restoration*. *September*. http://etheses.whiterose.ac.uk/24873/

Ezeani, I., Hepple, M., Onyenwe, I., &Enemuo, C. (2018a). Igbo diacritic restoration using embedding models. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Student Research Workshop*, *2018-Janua*, 54–60. https://doi.org/10.18653/v1/n18-4008

Ezeani, I., Hepple, M., Onyenwe, I., &Enemuo, C. (2018b). Transferred Embeddings for Igbo Similarity, Analogy and Diacritic Restoration Tasks. *COLING 2018 - 3rd Workshop on Semantic Deep Learning, SemDeep 2018 - Proceedings*, 30–38.

Hifny, Y. (2018). Hybrid LSTM/MaxEnt Networks for Arabic Syntactic Diacritics Restoration. *IEEE Signal Processing Letters*, *25*(10), 1515–1519. https://doi.org/10.1109/LSP.2018.2865098

Hifny, Y. (2019). Open Vocabulary Arabic Diacritics Restoration. *IEEE Signal Processing Letters*, *26*(10), 1421–1425. https://doi.org/10.1109/lsp.2019.2933721

Hifny, Y. (2021, June). Recent Advances in Arabic Syntactic Diacritics Restoration. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7768-7772). IEEE.

Hucko, A., &Lacko, P. (2018). Diacritics restoration using deep neural networks. *DISA 2018 - IEEE World Symposium on Digital Intelligence for Systems and Machines, Proceedings*, 195–200. https://doi.org/10.1109/DISA.2018.8490624

Hung, B. T. (2018). Vietnamese Diacritics Restoration Using Deep Learning Approach. *Proceedings of 2018 10th International Conference on Knowledge and Systems Engineering, KSE 2018*, 347–351. https://doi.org/10.1109/KSE.2018.8573427

Hung, B. T. (2019). Integrating diacritics restoration and question classification into Vietnamese question answering system. *Advances in Science, Technology and Engineering Systems*, *4*(5), 207–212. https://doi.org/10.25046/aj040526

Kishore, A., Kumar, A., Singh, K., Punia, M., &Hambir, Y. (2018). Heart Attack Prediction Using Deep Learning. *International Research Journal of Engineering and Technology (IRJET)*, *5*(4), 4420–4423.

Laki, L. J., & Yang, Z. G. (2020). Automatic diacritic restoration with transformer model based neural machine translation for east-central european languages. *CEUR Workshop Proceedings*, *2650*, 190–202.

Mariel, W. C. F., Mariyah, S., &Pramana, S. (2018). Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and naïve Bayes for Indonesian text. *Journal of Physics: Conference Series*, *971*(1). https://doi.org/10.1088/1742-6596/971/1/012049

Masmoudi, A., Mdhaffar, S., Sellami, R., &Belguith, L. H. (2019). Automatic diacritics restoration for Tunisian dialect. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *18*(3). https://doi.org/10.1145/3297278

Mihalcea, R. F. (2002). *Diacritics Restoration : Learning from Letters versus Learning from Words*. 339–348.

Náplava, J., Straka, M., Straňák, P., &Hajic, J. (2018, May). Diacritics restoration using neural networks. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Náplava, J., Straka, M., Straňák, P., &Hajič, J. (2019). Diacritics restoration using neural networks. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 1566–1573.

Nga, C. H., Thinh, N. K., Chang, P. C., & Wang, J. C. (2019, December). Deep learning based Vietnamese diacritics restoration. In *2019 IEEE international symposium on multimedia (ISM)* (pp. 331-3313). IEEE.

Nuţu, M., Lőrincz, B., & Stan, A. (2019, September). Deep learning for automatic diacritics restoration in romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 235-240). IEEE.

Oladipo, F. (2017). *The development of a Standard Yorùbá Diacritics Restoration System ` a development of a standard yorub digital text automatic diacritic By*. *May 2014*. https://doi.org/10.13140/RG.2.2.35584.12800

Orife, I. (2018). Attentive sequence-to-sequence learning for diacritic restoration of yorùbá language text. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2018-Septe*, 2848–2852. https://doi.org/10.21437/Interspeech.2018-42

Orife, I., Adelani, D. I., Fasubaa, T., Williamson, V., Oyewusi, W. F., Wahab, O., &Tubosun, K. (2020). *Improving Yor\ `ub\ 'a Diacritic Restoration*. 1–4. http://arxiv.org/abs/2003.10564

Rotimi, A. E. (2005). *A New Approach to Tone Mark Restoration in Standard YorùbáText : A Proposal*. *1*, 8–19.

Šanti, N., &Šnajder, J. (2009). *Automatic Diacritics Restoration in Croatian Texts*. 309–318.

Schlippe, T., & Vogel, S. (2014). *Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem Diacritization as a Machine Translation Problem and as a Sequence Labeling Problem*. *October*.

Schrumpf, C., Larson, M., &Eickeler, S. (2005). *Syllable-based Language Models in Speech Recognition for English Spoken Document Retrieval*. 196–205.

Wagacha, P. W., Pauw, G. De, &Getao, K. W. (2006). *˜ u ˜ using language-independent Developing an annotated corpus for G ˜ ıkuy machine learning techniques*.

Wagacha, P. W., Pauw, G. De, &Githinji, P. W. (2002). *A Grapheme-Based Approach for Accent Restoration in G ˜ ıkuy*. 1937–1940.

Yarowsky, D. (1863). *A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Text*. 1–14.

Yasser, H. (2021). *Yasser Hifny University of Helwan , Egypt*. 7768–7772.