The Effect of Domain and Ambiguity in Yorùbá Automatic Diacritic Restoration

Micheal Tomiwa Oyebamiji[1*], Jennifer Ogochukwu Ibekwe[2], Faith Ayomide Ajiboyede[3]

[1]Department of Data Science, Faculty of Computing, University of Ibadan, Ibadan, Oyo, Nigeria
[2]Department of Linguistics, Faculty of Arts, University of Uyo, Uyo, Akwa-Ibom, Nigeria
[3]Department of Computer Science, Faculty of Computing, University of Ibadan, Ibadan, Oyo, Nigeria
E-mail: [1*]oyebamijimicheal10@gmail.com, [2]jennifieribekwe50@gmail.com, [3]ajiboyedefaith@gmail.com

(Received: 21 Sep 2026, revised: 4 Nov 2026, 14 Nov 2026, accepted: 18 Nov 2026)

Abstract

In Yorùbá, diacritics are used to denote the meaning of a word. Without them, the same string of characters may mean different things. Efforts to automatically restore diacritics have mostly used test text from the same sources as the training data. To assess performance outside that range, four Automatic Diacritic Restoration (ADR) systems were evaluated on two datasets: 664 sentences from the West African Senior School Certificate Examination (WAEC) Yorùbá paper, and 603 sentences from MultiDiac, a dataset built around words with several valid diacritized forms. Inputs were created by stripping diacritics from the datasets, and outputs were scored with WER, CER, DER and WDER. For robustness, two baselines were added: one returning the input unchanged, the other giving each word its most frequent diacritized form in the Menyo-20k training corpus. Systems were compared using paired bootstrap tests, with MultiDiac scored on its held out dev split. Each reference word was classified by whether it appeared in the training corpus in the required diacritized form, a different form, or not at all. On DER, the BiLSTM ami (1.29M parameters) and mT5_base_yoruba_adr (582.4M) did not differ significantly on WAEC (0.4025 vs 0.4187) or MultiDiac (0.4349 vs 0.4273), while both outperformed the unigram baseline (0.5102 on WAEC, 0.4805 on MultiDiac). Errors on MultiDiac were higher despite better vocabulary coverage, because many of its words appeared in training mostly in a different form. These findings suggest that more varied training data, rather than larger models, may be the better route to improved restoration.

Keywords: Yorùbá, Automatic Diacritic Restoration, Domain Adaptation, Lexical Ambiguity, Low-resource NLP.

I. INTRODUCTION

Yorùbá is spoken by more than 50 million people, mainly in south-western Nigeria, Benin and Togo [1, 2]. It is a tonal language having three tones: low, mid and high, indicated by grave, optional macron and acute accent marks above vowels and syllabic nasals. There are three more characters with underdots to indicate particular sounds: ẹ, ọ and ṣ [2]. These marks are not stylistic in nature. The word owo, without any marks, is the same time money, trade, hand and broom, depending on the diacritics and their position. This is shown in Table 1. Without diacritics, the text in this type of language will be more difficult to read, and also more ambiguous in ways that can only be resolved by context.

These marks are not used in most digital spaces of Yorùbá text. News websites, social media and informal messages are mostly written without diacritics, either because the authors do not have the tools on their keyboards to insert them or because they have become an established practice over time [3, 4]. The consequence of this is that good quality training data for ADR models is generally only representative of a small subset of the contexts in which it is actually required to produce a correctly diacritised version of Yorùbá text.

Table 1. The Undiacritized Form Owo Resolves to Four Distinct Yorùbá Words Depending on Diacritization

Undiacritized	Diacritized Form	Meaning
owo	owó	money
owo	òwò	trade
owo	ọwọ́	hand
owo	ọwọ̀	broom

The domains of most available Yorùbá ADR training are based on the Menyo-20k corpus, which spans a fair amount of domains such as news, religious text, TED talks, proverbs, radio transcripts and conversational material [5]. It does not however represent examination language, literary criticism,

DOI: 10.34148/teknika.v15i3.1600
TEKNIKA, Volume 15(3), November 2026, pp. 583-596
ISSN 2549-8037, EISSN 2549-8045

CC BY SA

grammatical discourse, and text based around lexically ambiguous words. These are the cases where the limits of what the model has learned are apparent when it encounters these kinds of text. This paper intends to bring these limits to light, to accurately gauge them and to draw out what they mean for how the field should direct its resources.

Improvement in NLP is frequently expressed as a series of models that improve on a common test. For Yorùbá ADR, this workflow does not include an important question. The models available are evaluated with text from the same sources they were trained with. If they are removed from that distribution, the question of which model performs best will no longer be of interest than the question of what causes all of them to struggle at roughly the same point. This paper seeks to investigate that question.

The problem addressed here is that published results for Yorùbá ADR are measured on text from the training domains, so it is unclear whether the errors that remain on other text come from words never seen in training, from words seen in a different diacritized form, or from the limited size of the models. The objectives are to measure how far the performance of publicly available Yorùbá ADR models drops on two out-of-domain datasets, to relate the errors to the vocabulary and the diacritized forms present in the Menyo-20k training corpus, and to test whether differences between models are larger than the sampling variation in the evaluation sets. The answers matter for anyone who has to choose between collecting new diacritized text and training larger models, since the two options compete for the same limited resources in a low-resource setting.

The contributions are as follows. First, four public ADR models are evaluated on WAEC examination text and on MultiDiac, with a unigram lookup and an unrestored input as baselines. Second, reference words are divided into four coverage categories relative to Menyo-20k and error rates are reported for each category, which separates unseen words from seen words that need a less frequent diacritized form. Third, all scores come with 95% bootstrap confidence intervals, and paired bootstrap tests are used for model comparisons. Fourth, dataset statistics, the annotation procedure, checkpoint revisions and the software environment are documented so that the evaluation can be repeated.

Three transformer models, mT5_base_yoruba_adr, byt5-small-diacritizer-menyo and omowe-t5-small-diacritizer-all-und-full, all from "Davlan" on Hugging Face, and one character-level BiLSTM [6], ami-yoruba-diacritics from "kenny0bi" also on Hugging Face, are compared on the WAEC examination corpus and on MultiDiac [7]. The error metrics are supplemented with a vocabulary coverage analysis that shows how much of each evaluation corpus is found in the Menyo-20k training vocabulary, both in its exact diacritized form and as an undiacritized surface string. The combination gives a better picture than the error rates alone.

II. EXISTING WORK

The development of diacritics for Yorùbá NLP has been in progress for more than 10 years and the picture that has emerged is fairly consistent. Based on a distributional analysis of over a million Yorùbá words, Asubiaro [3] found that the absence of diacritics results in the loss of a significant amount of statistical information from any downstream model. Words that are phonologically and semantically distinct become the same string, and a model trained on the mixed-orthography text has a noisier input than it seems. Wells [4] demonstrated that the omission of diacritics is a pattern found in many languages in digital communication, and attributed to a number of factors including keyboard limitations and early computing conventions. In Yorùbá the consequences are more severe because of the tonal structure of the language, but the source of the problem is a shared one.

Downstream effects have been examined for various tasks. Adelani et al. [5] demonstrated the usefulness of diacritization in machine translation, where they employed Menyo-20k to show that the advantage of diacritization is related to the similarity between the training and test domains. When they are close, the effect is smaller. When the train and test domain are different, correct diacritization usually helps. When Yorùbá is the target language, they pointed out that it is always necessary to have the right diacritics. Akinlabi and Sanghal [8] extended this observation to commercial translation systems, and found that systems such as Google Translate can generate fluent-sounding Yorùbá, but with incorrect diacritics, a failure mode different to producing incorrect words. At the representation level, the results from Adewumi et al. [9] show that word embeddings trained on undiacritized Yorùbá can perform as well as those trained on diacritized Yorùbá on word similarity tasks, not because diacritics do not carry information, but because the pre-training data used to train the word embeddings is predominantly undiacritized, making the diacritized signal an unfamiliar one as opposed to a useful one.

Early research work on Yorùbá ADR was by Orife [10] who posed it as a sequence-to-sequence translation problem and obtained an error rate of less than 5% on a held-out test set. Orife et al. [11] followed that up with a study that revealed that the best lever for improving restoration quality is a combination of text from different domains. The JW300 corpus gave a 33% BLEU improvement over the previous baseline. Adelani et al. [5] studied the effect of domain and diacritics in Yorùbá-English neural machine translation. They trained an ADR system on the Yorùbá side of JW300 and Menyo-20k and achieved a BLEU of 87.0 on a held-out test set; the mT5_base_yoruba_adr model released alongside that work is one of the four models evaluated here. Olawole et al. [12] presented the YAD benchmark. The study pre-trained Oyo-T5 on Yorùbá text, and it was observed that models trained on a particular domain perform best on the same domain, and that larger models and more training data yield better performance. It is important to note that YAD is based on the Menyo-20k test split, therefore most published results of Yorùbá ADR have been carried out on the same distribution used for training. Ayoola and Olukanmi [13] tested four

systems on YAD, two of which were frontier LLM models, and concluded that the specialized mT5 model is slightly superior to LLM models on error rate measures, but also that it is faster, cheaper, and fully offline. They also reported that about 43% of all diacritic locations are mispredicted by the best performing system on its own test set, which gives some sense of how much room for improvement remains.

The ambiguity in Yorùbá has been studied from the linguistics side for some time. Adegoke-Elijah et al. [14] recorded disambiguation patterns in Yorùbá verbs, and demonstrated that the meaning of many verbs can be correctly disambiguated only from the semantic class of their argument. Akindele [15] solved the same problem by adopting the ontological annotation technique where ambiguous verbs are formalised in machine-readable format to aid NLP tools. Both works point to what the owo example in Table I shows: that undiacritized Yorùbá is not merely incomplete, it is structurally ambiguous in a way that no model can resolve by surface pattern matching alone. With this in mind, MultiDiac [7] was developed. Toyin et al. built sentences on the basis of base words that have multiple valid diacritized forms, with the help of native speakers to ensure that the intended reading was always recoverable from context. In the same paper, omowe-t5 and byT5 were compared with LLM baselines and it was noted that in some cases, the specialized models generate incomplete outputs. This is important because it can affect the interpretation of those results; a truncated hypothesis leads to an inflation of every error metric in a way that is not easily comparable to a complete but imperfectly diacritized response.

As a whole, African language NLP has been gradually improving, albeit limited by the availability of data. Adebara and Abdul-Mageed [16] surveyed the current status of NLP in African languages and reported that one of the major challenges is the lack of quality and diversity of training data. Inuwa-Dutse [17] carried out a survey on the Nigerian language NLP and found diacritics and tone modelling as the major challenges for Yorùbá, Hausa and Igbo. Jimoh et al. [18] reviewed the development of Yorùbá NLP over the past ten years and found that one of the primary challenges for reliable representations of Yorùbá is the diacritic inconsistency in the available text. The findings of the evaluation in this paper are in line with these observations.

Two gaps follow from the review of existing works. First, published Yorùbá ADR results are mostly reported on held-out test sets that come from the same distribution as the training data, for example the Menyo-20k-based YAD benchmark [12], so little is known about how the released models behave on text from other domains. Second, MultiDiac [7] provides sentences built around ambiguous words, but it was introduced to compare general-purpose language models with specialized ADR models, and to the best of the authors' knowledge the relation between its difficulty and what the specialized models saw in training has not been examined. This study addresses both gaps by evaluating four public ADR models on two datasets that lie outside the domains of Menyo-20k, and by relating the errors on each dataset to the diacritized forms of each word that appear in the Menyo-20k training corpus.

III. RESEARCH METHODOLOGY

A. Methodology Overview

The research design is summarized in Figure 1, where the two out-of-domain test sets are stripped of diacritics. The undiacritized sentences are passed to the four public models and the two baselines. The restored text is then scored against the original diacritized sentences. Four error metrics are computed for every system. The Menyo-20k training corpus is used to build the Unigram baseline and to support the coverage analysis, which relates the errors to the words seen in training. Paired bootstrap tests then check whether the differences between systems are larger than chance. Together, these results show whether more varied training data or larger models is the better route to improved restoration.

B. Datasets

WAEC corpus. The West African Senior School Certificate Examination (WASSCE) Yorùbá language paper is conducted every year by West African Examinations Council (WAEC). Yorùbá WAEC questions were collected from two sources: the first, referred to as paper 1, is the printed version of the exam while the second, referred to as paper 2, was sourced directly from the e-learning portal. Paper 1 covers eight examination years (2016 to 2022 and 2024). The printed pages were converted to text by prompting DeepSeek-V4-Pro-0813 to extract the text from the attached images. The output from the model contains noise, such as an underdot that was not added or a high tone that was omitted. The outputs were therefore reviewed and corrected using the online Yorùbá keyboard of Lexilogos [19]: the text was pasted into the keyboard page, the missing or wrong marks were fixed, and the corrected text was copied back. This keyboard was preferred because tones and underdots are available as single clicks, whereas other keyboards require pressing the Ctrl or Shift key. Paper 2 was taken from the portal as text and was not passed through OCR. Paper 1 passages were split on full stops while the corresponding questions and answer options were used without further splitting. Single-word entries were excluded from both papers to give room for context disambiguation. The final corpus contains 664 samples.

MultiDiac. MultiDiac [7] is a dataset constructed around base words selected for their diacritic ambiguity. For the Yorùbá portion, approximately 71 base words were identified and used to build sentences in collaboration with three native speakers holding advanced academic qualifications in the language. The public release on Hugging Face has a train split of 562 samples and a dev split of 41 samples. The paper that introduced the dataset also describes a gold-standard test split of 101 samples, which was checked by an independent native speaker and the lead annotator [7], but this split is not part of the public release, and the independent check is described for the test split only. The train and dev splits were combined into 603 samples because, to the best of the authors' knowledge, none of the evaluated models was trained on MultiDiac, and because 41 samples alone are too few for stable estimates. Results on the dev split are reported separately so that the held-out part of the release can be inspected on its own.

DOI: 10.34148/teknika.v15i3.1600

Undiacritized inputs. For both datasets, the undiacritized versions were created by mapping the three underdot characters (ẹ, ọ, ṣ) to (e, o, s), with their uppercase counterparts handled in the same way, and then removing all other combining marks from the strings by using Unicode NFD decomposition.

Dataset statistics. Table 2 gives the number of samples, word tokens and word types for each source. Tokens are lowercased, NFC-normalized alphabetic strings with diacritics kept, and a type is a distinct diacritized token. WAEC has 6,103 tokens and 1,475 types over 664 samples. Paper 2 supplies 250 samples and 2,234 tokens, and each of the eight paper 1 years supplies between 34 and 63 samples. MultiDiac has 5,575 tokens and 1,452 types over 603 samples, of which 562 come from the train split and 41 from the dev split. Samples are short in both datasets, with about 9 tokens on average.

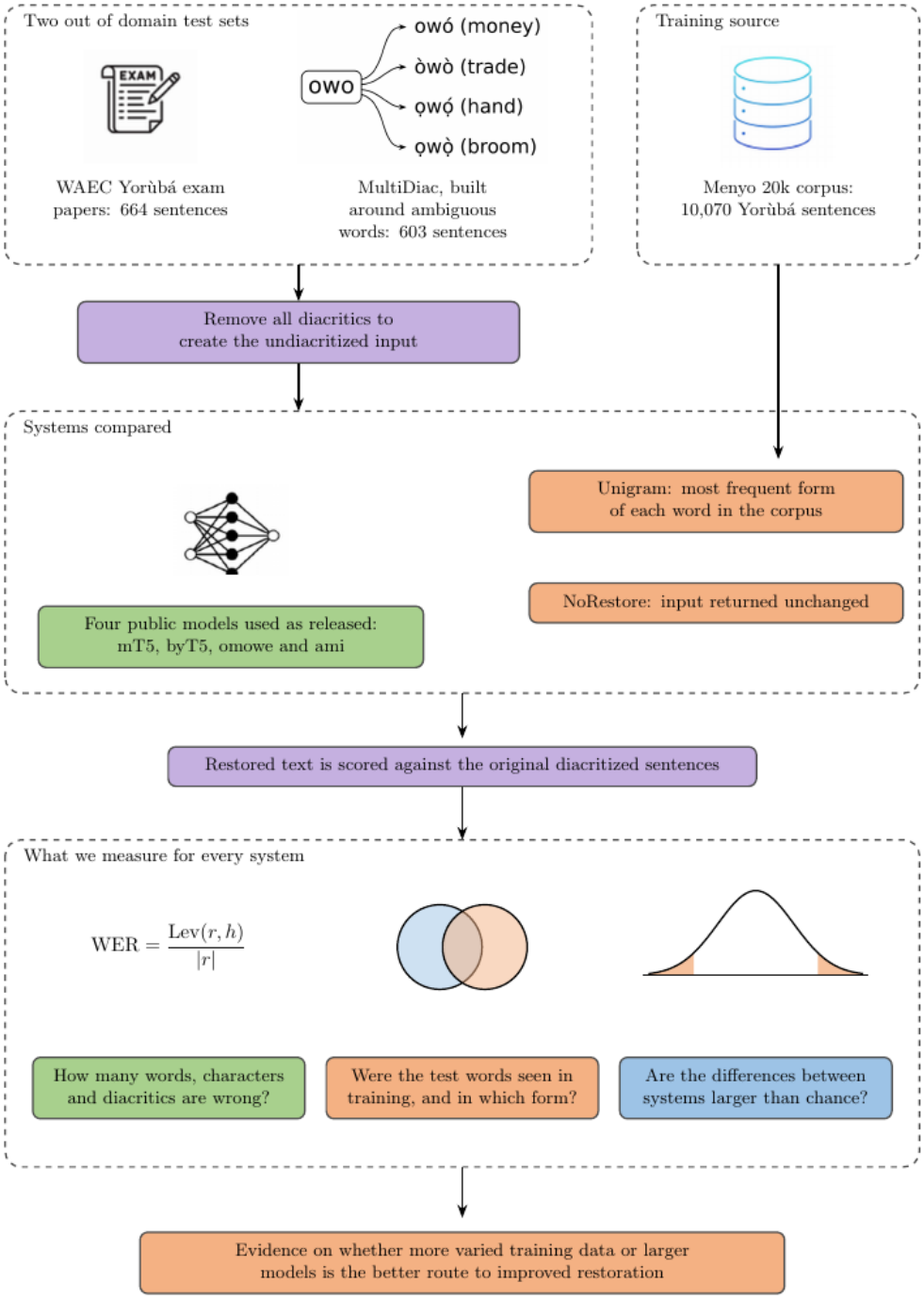


Figure 1. Architecture of the Study Which Shows Progression From the Evaluation Datasets and the Menyo-20k Training Corpus to the Evidence on Training Data Versus Model Size

Table 2. Number of Samples, Tokens, and Types for Each Dataset Source.

Dataset	Source	Samples	Tokens	Types
WAEC	Paper 1, 2016	60	438	202
WAEC	Paper 1, 2017	50	525	232
WAEC	Paper 1, 2018	55	431	189
WAEC	Paper 1, 2019	46	430	218
WAEC	Paper 1, 2020	48	495	226
WAEC	Paper 1, 2021	58	539	238
WAEC	Paper 1, 2022	34	384	193
WAEC	Paper 1, 2024	63	627	239
WAEC	Paper 2	250	2,234	705
WAEC	**Total**	**664**	**6,103**	**1,475**
MultiDiac	Train split	562	5,196	1,403
MultiDiac	Dev split	41	379	233
MultiDiac	**Total**	**603**	**5,575**	**1,452**

Filtering and data checks. Table 3 reports the checks that were applied. The 221 single-word WAEC entries were removed, while no MultiDiac sample was a single word. Exact duplicates after NFC normalization and lowercasing were found in 11 WAEC samples and 4 MultiDiac samples and were kept, since each occurrence is a separate item in its source. None of the evaluation sentences occurs among the Menyo-20k training sentences, so there is no sentence-level overlap with that corpus. A small number of samples mix precomposed and decomposed characters in the same sentence (19 in WAEC and 13 in MultiDiac), and one WAEC sample is fully decomposed (NFD).

Table 3. Filtering, Duplicate, Overlap, and Unicode Form Checks.

Check	WAEC	MultiDiac	Menyo-20k train
Single-word entries excluded	221	0	n/a
Duplicate samples	11	4	n/a
Sentences found in Menyo-20k train	0	0	n/a
Samples in NFC	641	590	10,042
Samples with mixed forms	19	13	0
Samples in NFD	1	0	0
Samples without composable marks	3	0	28

Annotation quality. For WAEC paper 1, the reference diacritics are the result of OCR followed by manual correction by the authors, with no second annotator, so no agreement score can be reported. Residual errors are therefore possible, and a missed underdot or tone mark in a reference counts against a model that outputs the correct form. For MultiDiac, the sentences were written by the annotators described in [7]. Table 4 shows three randomly selected samples from each dataset (random seed 42) in diacritized and undiacritized form. The WAEC samples are examination instructions and short phrases about Yorùbá literature and grammar, while the MultiDiac samples are short everyday sentences built around ambiguous words.

Table 4. Randomly Selected Samples in Diacritized and Undiacritized Form (Index in the Dataset in Parentheses).

Dataset	Form	Text
WAEC (434)	Diacritized	Nínú “Orikì Ìran Olúfẹ̀”, sọ ìtàn tí ó rọ̀ mọ́ gbólóhùn tí ó sọ pé “Adé Òrùrù la gbà fún Aláàfin”.
WAEC (434)	Undiacritized	Ninu “Oriki Iran Olufe”, so itan ti o ro mo gbolohun ti o so pe “Ade Oruru la gba fun Alaafin”.
WAEC (59)	Diacritized	Iṣẹ́ àgbẹ̀ àǹfààní
WAEC (59)	Undiacritized	Ise agbe anfaani
WAEC (513)	Diacritized	Kọ àpẹẹrẹ òrọ̀-asopọ̀ márùn-ún kí o sì lo ọ̀kọ̀ọ̀kan wọn nínú gbólóhùn.
WAEC (513)	Undiacritized	Ko apeere oro-asopo marun-un ki o si lo okookan won ninu gbolohun.
MultiDiac (51)	Diacritized	Inú igbó ní wón sin òkú ẹni ti o bá ní iké sí
MultiDiac (51)	Undiacritized	Inu igbo ni won sin oku eni ti o ba ni ike si
MultiDiac (420)	Diacritized	Orí igún kò ṣàdédé pá, ó ní ìtàn tó rọ̀ mọ
MultiDiac (420)	Undiacritized	Ori igun ko sadede pa, o ni itan to ro mo
MultiDiac (516)	Diacritized	Bàbá mí ra ọkọ̀ ayọ́kẹ́lẹ́ ní àná
MultiDiac (516)	Undiacritized	Baba mi ra oko ayokele ni ana

C. Models

Table 5 describes the four models evaluated and the two baselines, and the short names used in the rest of the paper. The four models can be found on HuggingFace. The first model is mT5_base_yoruba_adr [5, 20] which was fine-tuned on JW300 and the Menyo-20k training and development splits. The second is byt5-small-diacritizer-menyo [21] and the third is omowe-t5-small-diacritizer-all-und-full [22] both of which were developed as part of the YAD project [12]. The byt5 model is a byte-level T5 variant [23] fine-tuned on the YAD training split. The omowe model is a small T5 fine-tuned on a broader combination of the Yorùbá Bible, JW300, and Menyo-20k. The fourth model is named ami, a 1.29M-parameter character BiLSTM trained on the Yorùbá side of the Menyo-20k training split plus the diacritized portion of the Yorùbá Text C3 corpus [6], filtered for diacritic density and de-duplicated against the evaluation splits. Parameter counts were obtained by counting the parameters of the loaded checkpoints. The models differ in architecture, tokenization, pretraining and

DOI: 10.34148/teknika.v15i3.1600

fine-tuning data, so comparisons between them are observational.

Two baselines were added. NoRestore passes the undiacritized input through unchanged, which gives the error rate of doing nothing, and by construction its DER and WDER are 1. Unigram is a context-free lookup built from the Menyo-20k training sentences: each word of the input is replaced by the most frequent diacritized form seen for that bare word, the first letter is capitalized when the input word is capitalized, and words absent from the training data are left unchanged. Unigram therefore uses the training vocabulary but no sentence context, which makes it a direct measure of how much of a score can be reached from vocabulary alone.

Table 5. Systems Evaluated in This Study.

Short Name	Architecture	Para meters	Training Data
mT5	mT5-base seq2seq	582.4M	JW300, Menyo-20k train and dev
byT5	ByT5-small, byte-level seq2seq	299.6M	YAD training split
omowe	T5-small seq2seq	76.9M	Yorùbá Bible, JW300, Menyo-20k
ami	Character BiLSTM	1.29M	Menyo-20k train, Yorùbá Text C3
Unigram	Lookup of most frequent form	n/a	Menyo-20k train
NoRestore	Identity	n/a	None

D. Menyo-20k Training Corpus

Menyo-20k [5] is a multi-domain English-Yorùbá parallel corpus. Table 6 shows its domain composition, which forms the primary training base for all four models. Only the Yorùbá side of the training split (10,070 sentences) was used here. It was fetched from the public repository of the corpus (uds-lsv/menyo-20k_MT, file train.tsv), and the digest of the sentences is given in Table 7. The dev and test splits were not used. The training sentences serve two purposes: they are the source of the vocabulary and form frequencies in the coverage analysis, and they are the source of the Unigram baseline. Each sentence was NFC-normalized, lowercased and split into alphabetic tokens (Latin letters, accented Latin letters and combining marks). For every bare word, obtained by removing all diacritics, the count of each diacritized form was stored, and the most frequent form was taken as its majority form.

Figure 2 shows the flowchart of the research conducted. The process starts with the WAEC, MultiDiac and Menyo-20k data. Diacritics are removed from the test sentences, and the four models and two baselines restore them. WER, CER, DER and WDER are then computed, and the errors are related to the Menyo-20k coverage. The last step is decided by the paired bootstrap test: when the difference between systems is significant, the systems are ranked, and otherwise no clear difference is reported. The subsections that follow describe each step of Figure 2.

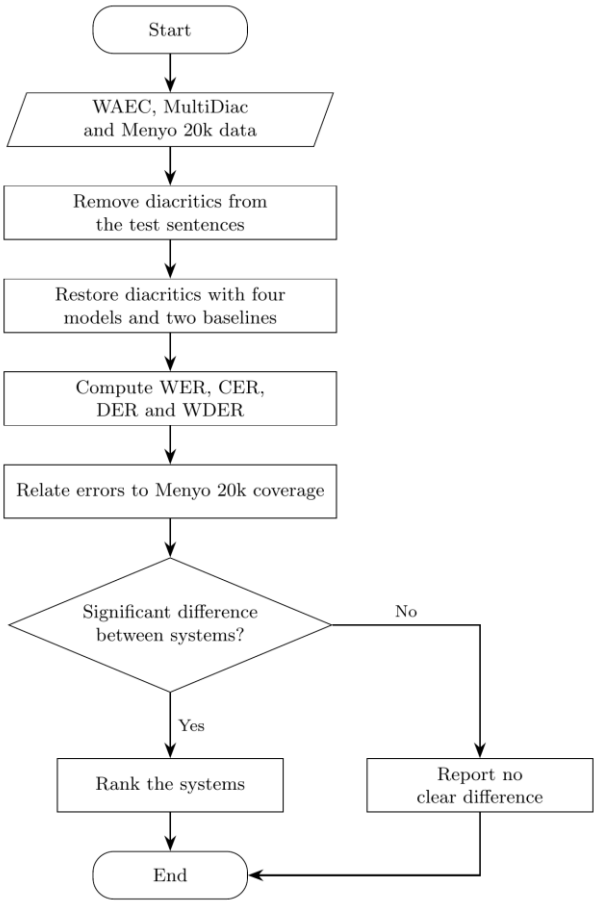


Figure 2. Flowchart of the Research Conducted

Table 6. Domain Composition of Menyo-20k [5].

Source	Training	Dev	Test
Global Voices News	1,487	1,391	1,388
Out of His Mind Book	0	1,006	1,008
TED Talks	507	438	2,000
ICT/digital sentences	0	312	273
Yorùbá Proverbs	2,200	250	250
VON News	1,714	0	0
JW News	3,508	0	0
Movie Transcript	774	0	0
Various domains	687	0	0
Kolibri Tech sentences	356	0	0
Radio Broadcast	258	0	0
Creative Commons	193	0	0
UDHR Translation	100	0	0
Total	**10,070**	**3,397**	**6,633**

E. Inference Settings

All models were used as released, without further training, so no training hyperparameters apply and the settings below are inference settings. The transformer models all used batched seq2seq generation through the generate method of the

Transformers library [24] with beam search (num_beams=4, early_stopping=True, max_new_tokens=512) and no sampling. Inputs were tokenized with the tokenizer shipped with each checkpoint (a SentencePiece tokenizer for mT5, a byte-level tokenizer for byT5 and the T5 tokenizer for omowe), padded to the longest item in the batch and truncated at 512 tokens. The ami model handled text processing by the native restore interface, processing one sentence at a time. All experiments were performed on the Kaggle platform using an NVIDIA Tesla T4 GPU, and a batch size of 32 for the transformer models and 64 for ami. Table VII lists the library versions, the exact revision of every checkpoint and of the MultiDiac release, the seed and the decoding configuration. The seed (42) fixes the bootstrap resamples, and since decoding uses no sampling no seed applies to the predictions. Each model was run once on each dataset, and predictions were saved after every batch so that a run could be resumed.

The experiments were run again for this revision with the versions in Table 7. The scores of mT5 and ami are identical to those of the first submission, while the scores of byT5 and omowe differ (for example, the WER of byT5 on WAEC is 0.5354 here against 0.6893 earlier). The cause was not isolated, and the two models appear to be more sensitive to the runtime environment than mT5 and ami, which should be kept in mind when their scores are read. The WAEC data are hosted in the Kaggle dataset oyebamijimicheal/effect-of-domain-and-ambiguity-in-yoruba-adr-data, and the evaluation notebook, the per-sentence predictions and the score files are released with it.

Table 7. Software, Checkpoints, and Settings Used for the Evaluation.

Item	Value
Software	Python 3.13.16; PyTorch 2.11.0+cu130; Transformers 5.20.2; Datasets 4.8.5; NumPy 2.1.3; pandas 2.3.3
Hardware	NVIDIA Tesla T4 GPU on the Kaggle platform
Seed	42 (NumPy and PyTorch); 2,000 bootstrap resamples
Decoding	model.generate with num_beams=4, early_stopping=True, max_new_tokens=512, no sampling (mT5, byT5, omowe)
Input handling	Tokenizer of each checkpoint, padding to the longest item in the batch, truncation at 512 tokens
Batch size	32 (mT5, byT5, omowe); 64 for ami, which processes one sentence at a time
Checkpoint, mT5	Davlan/mT5_base_yoruba_adr, revision 9a6c4a7ba523ea0c6f6e5acc67592b454204e384
Checkpoint, byT5	Davlan/byt5-small-diacritizer-menyo, revision 6ecaa2d0e0e9b75b1246186567f5c3a3e9d25c96
Checkpoint, omowe	Davlan/omowe-t5-small-diacritizer-all-und-full, revision 89f8bffc723d92007297227ec34c81f94aa55c9f
Checkpoint, ami	kenny0bi/ami-yoruba-diacritics, revision 06b1e4f12f5c68df8288f237dde4f817a0b7137c
MultiDiac release	herwoww/MultiDiac, revision 9cb4e0a4e6a38588d7ee2431221040ae25f9e2ae
Menyo-20k training text	SHA-256 of the 10,070 Yorùbá sentences joined by line breaks: 1aeddfd59fb26900905825d7a2476610280090d4b2234aaa2865c3263ef1a7ff
Unicode handling	Undiacritized inputs: NFD with marks removed. Coverage analysis: NFC and lowercase (Unicode database 15.1.0). Scoring: strings compared as stored, with no normalization

F. Evaluation Metrics

Four error-rate metrics are reported, all based on the Levenshtein edit distance [25] which is the number of substitutions, deletions and insertions required to transform one sequence into another. All four are computed over the whole dataset: errors and reference lengths are summed over all sentences before the division. For the k-th reference sentence r_k and its hypothesis h_k, let w(.) be the sequence of whitespace-separated words, c(.) the sequence of characters and Lev(.,.) the edit distance.

WER (Word Error Rate) divides the total word-level edit distance between hypothesis and reference by the total number of words in the reference [26], as in Equation 1. A WER of 0.20 means that the number of edits equals one fifth of the number of reference words.

$$\mathrm{WER} = \frac{\sum_k \mathrm{Lev}\left(\mathrm{w}(r_k),\mathrm{w}(h_k)\right)}{\sum_k |\mathrm{w}(r_k)|} \quad (1)$$

CER (Character Error Rate) applies the same calculation at the character level [27], as in Equation 2. Because Yorùbá diacritics are combining characters attached to base letters, CER captures errors that WER would only count once at the word level. This is a more sensitive measure for this task.

$$\mathrm{CER} = \frac{\sum_k \mathrm{Lev}\left(\mathrm{c}(r_k),\mathrm{c}(h_k)\right)}{\sum_k |\mathrm{c}(r_k)|} \quad (2)$$

DER (Diacritization Error Rate) restricts attention to positions in the reference that carry a diacritic, including combining marks and the underdot characters {ẹ, Ẹ, ọ, Ọ, ṣ, Ṣ}, and measures the proportion of those positions where the hypothesis differs from the reference [13, 28], as in Equation 3. Here D(r_k) is the set of character positions of r_k that hold such a mark, I(.) equals 1 when its condition holds and 0 otherwise, and a position beyond the end of the hypothesis counts as an error. A model that gets every base character right but places the wrong tone on every vowel would score a DER of 1.0. Positions are compared by index, without alignment, so

DOI: 10.34148/teknika.v15i3.1600
TEKNIKA, Volume 15(3), November 2026, pp. 583-596
ISSN 2549-8037, EISSN 2549-8045

an insertion or deletion early in a sentence shifts every later position.

$$\mathrm{DER} = \frac{\sum_k \sum_{i \in \mathrm{D}(r_k)} \mathrm{I}(h_k[i] \neq r_k[i])}{\sum_k |\mathrm{D}(r_k)|} \quad (3)$$

WDER (Word Diacritization Error Rate) counts the proportion of words in the reference that contain at least one diacritic where the hypothesis differs in any diacritic position [13, 29], as in Equation 4. Here W(r_k) is the set of word positions of r_k that contain a mark, and w_j(.) is the j-th word of a sentence. Unlike DER, it treats a word as either right or wrong regardless of how many individual marks within it are incorrect, and words are also compared by index. Scoring is done on the strings as stored, with no Unicode normalization.

$$\mathrm{WDER} = \frac{\sum_k \sum_{j \in \mathrm{W}(r_k)} \mathrm{I}\left(w_j(h_k) \neq w_j(r_k)\right)}{\sum_k |\mathrm{W}(r_k)|} \quad (4)$$

G. Coverage Categories

For the coverage analysis, every word token of the references is placed in exactly one of four categories relative to the Menyo-20k training sentences. Words are lowercased and NFC-normalized, and the bare word is the token with all diacritics removed. Seen, majority form: the exact diacritized token occurs in Menyo-20k and is the most frequent form of its bare word. Seen, minority form: the exact token occurs, but another form of the bare word is more frequent. Seen, other form: the bare word occurs in Menyo-20k, but never with the diacritics required by the reference. Unseen word: the bare word does not occur in Menyo-20k in any form, which is the pure OOV case. The first two categories together give the exact token coverage, and the first three give the surface token coverage. Types are counted over distinct diacritized tokens.

Error rates per category are computed by aligning the hypothesis words to the reference words on their undiacritized form with the SequenceMatcher of the Python difflib module. A reference word with no aligned hypothesis word, or whose aligned word differs from it, is counted as an error. For reference words that are not in the majority form (minority and other form), the share of cases in which the system output the majority form of Menyo-20k is also reported.

H. Statistical Analysis

The bootstrap resamples whole sentences rather than individual words or characters. This was done to ensure that all the words in a sentence stay together in each resample. For each dataset, 2,000 bootstrap resamples of the sentence indices were drawn with replacement (seed 42), and the same resamples were used for all systems, which makes the comparisons paired [30], [31]. In each resample every metric is recomputed as the ratio of summed errors to summed reference length, and the 95% confidence interval is given by the 2.5th and 97.5th percentiles. For a pair of systems, Δ^* is the difference between their metric values in one resample, the interval of Δ^* is reported, and the two-sided p-value is computed as in Equation 5. Three pairs were tested: mT5 and ami, mT5 and Unigram, and ami and Unigram. With 2,000 resamples the smallest nonzero p-value is 0.0005, so smaller values are reported as $p < 0.001$. No correction for multiple comparisons was applied, so the p-values are read as descriptive. Intervals were not computed for the dev split of MultiDiac because of its size.

$$p = \min\left(1 \middle| 2\min\left(\mathrm{P}(\Delta^* \leq 0), \mathrm{P}(\Delta^* \geq 0)\right)\right) \quad (5)$$

IV. RESULTS AND DISCUSSION

A. WAEC Results

The results of the evaluation on WAEC corpus are presented in Table 8. mT5 and ami are close across all four metrics. ami has the lowest CER (0.0807 vs. 0.1026), DER (0.4025 vs. 0.4187) and WDER (0.2109 vs. 0.2174), while mT5 has the lowest WER (0.2031 vs. 0.2063). The paired tests in Table XIV show that only the CER difference is reliable (mT5 minus ami +0.0219, 95% CI [0.0022, 0.0532], p = 0.010). The intervals of the WER, DER and WDER differences include zero (p = 0.694, 0.358 and 0.758), so the two models cannot be separated on those metrics. Both are well ahead of omowe and byT5. The byT5 model mispredicted about 93% of all diacritic positions in this dataset (DER 0.9262) and omowe about 72% (DER 0.7227). Table 9 shows one sample for the two best models.

Table 8. Evaluation Results on WAEC (664 Samples). Lower is Better for All Metrics.

System	WER	CER	DER	WDER
mT5	0.2031	0.1026	0.4187	0.2174
byT5	0.5354	0.3548	0.9262	0.8974
omowe	0.3744	0.2878	0.7227	0.5794
ami	0.2063	0.0807	0.4025	0.2109
Unigram	0.3170	0.1305	0.5102	0.2865
NoRestore	0.8695	0.3750	1.0000	1.0000

Table 9. Sample Outputs on WAEC for the Two Best Models.

System	Text
Reference	Ní ọjọ́ ìṣílé-iṣẹ́ yìí, ìlú kún fófó ayé gbọ́, ọ̀run sì mọ̀
mT5	Ní ọjọ́ ìṣílẹ̀-iṣẹ́ yìí, ìlú kún fọ́fọ́ ayé gbó, ọ̀run sì mọ́
ami	Ní ọjọ́ ìsílẹ̀-iṣẹ́ yìí, ìlú kún fọ́fọ́ ayé gbọ́, ọ̀run sì mọ̀

In this sample both models restore the underdots and tones of most words, and they differ from the reference at ìṣílé-iṣẹ́ and fófó, where both output forms with underdots (ẹ in ìṣílẹ̀ and ọ́fọ́ in fọ́fọ́). Whether the reference or the model is right at these positions needs a check by a second annotator, as discussed under the limitations. The two models differ from each other in the final words, where mT5 outputs gbó and mọ́ and ami outputs gbọ́ and mọ̀, the same forms as the reference.

CC BY SA

B. MultiDiac Results

The results on MultiDiac are given in Table 10. The WER of every model is higher on MultiDiac than on WAEC, and for the two best models the CER and WDER are higher as well (mT5 CER 0.1275 vs. 0.1026, ami WDER 0.2970 vs. 0.2109). DER is about the same for mT5 (0.4273 vs. 0.4187) and higher for ami (0.4349 vs. 0.4025), while byT5 and omowe have a lower DER on MultiDiac than on WAEC (0.8596 vs. 0.9262 and 0.6778 vs. 0.7227). A harder word-level result is expected based on the nature of the data, but it does not hold for every metric. Unlike on WAEC, mT5 is ahead of ami on all four metrics on MultiDiac, and the paired test shows that the advantage is reliable for WER (-0.0266), CER (-0.0059) and WDER (-0.0190) but not for DER (-0.0076, 95% CI [-0.0293, 0.0143], p = 0.501). What is happening at the sentence level is illustrated in Table 11.

Table 10. Evaluation Results on Multidiac (603 Samples). Lower is Better for All Metrics.

System	WER	CER	DER	WDER
mT5	0.3229	0.1275	0.4273	0.2780
byT5	0.5680	0.3223	0.8596	0.8219
omowe	0.5519	0.3826	0.6778	0.5772
ami	0.3495	0.1334	0.4349	0.2970
Unigram	0.3975	0.1500	0.4805	0.3295
NoRestore	0.7448	0.3078	1.0000	1.0000

Table 11. Sample Outputs on MultiDiac.

Example	System	Text
Example 1	**Reference**	Ówo tí ó sọ mí ní igi imú ni kò jẹ́ kí n mi dáadáa
Example 1	mT5	Owó tí ó sọ mí ní igi imú ni kò jẹ́ kí n mí dáadáa
Example 1	ami	Owó tí ó sọ mí ní igi imú ni kò jẹ́ kí n mí dáadáa
Example 2	**Reference**	Abọ́ ayọ́ ni ọmọ náà fi n jẹun
Example 2	mT5	Abo ayọ̀ ni ọmọ náà fi ń jẹun
Example 2	ami	Àbọ́ ayọ̀ ni ọmọ náà fi ń jẹun
Example 3	**Reference**	kí ni wọn mú oyún wúndíá se ní òkò
Example 3	mT5	kí ni wọ́n mú oyún wúńdíá ṣẹ ní oko
Example 3	ami	kí ni wọ́n mú oyún wúńdíá ṣe ní ọkọ̀

In the first example, the first word in the reference is Ówo (Boil) which is a less frequent tonal variant. "A boil is a painful, pus-filled lump which appears under the skin". Both mT5 and ami produce Owó (money), the form that appears most often for this base string in training data. The sentence context makes the intended reading clear to a human reader, but neither model uses that contextual information. The two models also output mí for the last verb where the reference has mi, and their outputs are identical, so they receive the same scores (WER 0.1333, CER 0.0600, DER 0.0769, WDER 0.0909). In the second example, the first word Abọ́ has a particular tonal form that is not captured by either model. mT5 dropped the diacritic entirely in this case while ami placed the wrong one. Both output ayọ̀ where the reference has ayọ́. The scores of the two models are the same for WER (0.3750) and WDER (0.4000) but very different for DER (1.0000 for mT5 against 0.1111 for ami). The gap is mostly a product of index-based comparison: mT5 turns the two characters ọ́ into the single character o, which shifts every later position, whereas ami keeps the string length. This is one reason why DER differences between models should be read together with CER and WER.

The third example shows that the models handle the interior of a sentence reasonably but disagree on the final word which is òkò (A town in Oyo state). ami outputs ọkọ̀ (vehicle) and mT5 outputs oko (farm). The two outputs receive the same WER (0.4444), CER (0.1765) and DER (0.8000). The MultiDiac dataset was designed to expose these types of cases. The reference sentences also do not mark the syllabic nasal in kí n mi and fi n jẹun, while the models output ń in the second example, so marking conventions are not fully consistent between references and models.

Because the original test split is not public, Table 12 reports the dev split, which is the only held-out split in the public release. The order of the systems is the same as on the full set: mT5 has the lowest error on all four metrics (WER 0.3182, CER 0.1234, DER 0.3842, WDER 0.2571), followed by ami (0.3342, 0.1271, 0.4337, 0.2643). With 41 samples and 379 tokens no interval is reported, and the dev split is read as a consistency check on the full-set results rather than as a separate result.

Table 12. Evaluation Results on the Multidiac Dev Split (41 Samples). Lower is Better for All Metrics.

System	WER	CER	DER	WDER
mT5	0.3182	0.1234	0.3842	0.2571
byT5	0.5615	0.3056	0.8535	0.8393
omowe	0.4465	0.2711	0.7050	0.5821
ami	0.3342	0.1271	0.4337	0.2643
Unigram	0.3690	0.1340	0.4495	0.2929
NoRestore	0.7487	0.3162	1.0000	1.0000

C. Baselines and Confidence Intervals

Table 13 lists every score with its 95% bootstrap interval and Table 14 gives the paired tests. Against the unigram lookup, mT5 and ami reduce DER by 0.0915 and 0.1077 on WAEC and by 0.0532 and 0.0456 on MultiDiac, and the intervals of all four differences exclude zero. The trained models therefore do better than vocabulary alone, but the margin is modest: a context-free lookup of the most frequent form already reaches a DER of 0.5102 on WAEC and 0.4805 on MultiDiac, and its WER (0.3170 and 0.3975) is 11.4 and 7.5 points behind mT5. For CER on WAEC, the advantage of mT5 over Unigram is not reliable (-0.0280, 95% CI [-0.0492, 0.0039], p = 0.075), while that of ami is (-0.0499). The unrestored input has a WER of 0.8695 on WAEC and 0.7448 on MultiDiac and a CER of 0.3750 and 0.3078, and its DER and WDER equal 1 by construction.

byT5 and omowe score worse than the unigram lookup on both datasets. Their WER and DER intervals do not overlap with those of Unigram (for example DER 0.9262 and 0.7227

against 0.5102 on WAEC), although these pairs were not tested formally. The intervals of omowe on MultiDiac are very wide for WER (0.4405 to 0.7587) and CER (0.2776 to 0.5789), which suggests that a few sentences with very long outputs dominate its score. Intervals of the DER of mT5 and ami are about 0.05 wide on each side, so differences smaller than that cannot be separated with datasets of this size.

Table 13. Scores With 95% Bootstrap Confidence Intervals (2,000 Paired Resamples of Sentences).

Dataset	System	WER	CER	DER	WDER
WAEC	mT5	0.2031 [0.1759, 0.2382]	0.1026 [0.0798, 0.1355]	0.4187 [0.3729, 0.4663]	0.2174 [0.1876, 0.2548]
WAEC	byT5	0.5354 [0.5102, 0.5657]	0.3548 [0.3272, 0.3911]	0.9262 [0.9134, 0.9376]	0.8974 [0.8737, 0.9176]
WAEC	omowe	0.3744 [0.3347, 0.4191]	0.2878 [0.2549, 0.3237]	0.7227 [0.6879, 0.7562]	0.5794 [0.5321, 0.6265]
WAEC	ami	0.2063 [0.1910, 0.2231]	0.0807 [0.0729, 0.0890]	0.4025 [0.3628, 0.4433]	0.2109 [0.1937, 0.2292]
WAEC	Unigram	0.3170 [0.3019, 0.3329]	0.1305 [0.1233, 0.1386]	0.5102 [0.4720, 0.5471]	0.2865 [0.2693, 0.3041]
WAEC	NoRestore	0.8695 [0.8611, 0.8778]	0.3750 [0.3700, 0.3802]	1.0000 [1.0000, 1.0000]	1.0000 [1.0000, 1.0000]
MultiDiac	mT5	0.3229 [0.3075, 0.3379]	0.1275 [0.1208, 0.1342]	0.4273 [0.3981, 0.4570]	0.2780 [0.2623, 0.2928]
MultiDiac	byT5	0.5680 [0.5519, 0.5830]	0.3223 [0.3085, 0.3349]	0.8596 [0.8376, 0.8790]	0.8219 [0.7968, 0.8452]
MultiDiac	omowe	0.5519 [0.4405, 0.7587]	0.3826 [0.2776, 0.5789]	0.6778 [0.6459, 0.7100]	0.5772 [0.5451, 0.6114]
MultiDiac	ami	0.3495 [0.3343, 0.3649]	0.1334 [0.1266, 0.1400]	0.4349 [0.4062, 0.4652]	0.2970 [0.2817, 0.3120]
MultiDiac	Unigram	0.3975 [0.3824, 0.4129]	0.1500 [0.1434, 0.1571]	0.4805 [0.4527, 0.5082]	0.3295 [0.3142, 0.3449]
MultiDiac	NoRestore	0.7448 [0.7310, 0.7569]	0.3078 [0.3012, 0.3140]	1.0000 [1.0000, 1.0000]	1.0000 [1.0000, 1.0000]

Table 14. Paired Bootstrap Tests for mT5, Ami and the Unigram Baseline. Differences are the First System Minus the Second, so a Negative Value Means That the First System Has the Lower Error.

Dataset	Comparison	Metric	Difference	95% CI of difference	p
WAEC	mT5 - ami	WER	-0.0032	[-0.0219, +0.0244]	0.694
WAEC	mT5 - ami	CER	+0.0219	[+0.0022, +0.0532]	0.010
WAEC	mT5 - ami	DER	+0.0162	[-0.0196, +0.0520]	0.358
WAEC	mT5 - ami	WDER	+0.0065	[-0.0141, +0.0366]	0.758
WAEC	mT5 - Unigram	WER	-0.1139	[-0.1379, -0.0828]	<0.001
WAEC	mT5 - Unigram	CER	-0.0280	[-0.0492, +0.0039]	0.075
WAEC	mT5 - Unigram	DER	-0.0915	[-0.1243, -0.0577]	<0.001
WAEC	mT5 - Unigram	WDER	-0.0691	[-0.0929, -0.0386]	<0.001
WAEC	ami - Unigram	WER	-0.1108	[-0.1229, -0.1003]	<0.001
WAEC	ami - Unigram	CER	-0.0499	[-0.0548, -0.0452]	<0.001
WAEC	ami - Unigram	DER	-0.1077	[-0.1390, -0.0773]	<0.001
WAEC	ami - Unigram	WDER	-0.0756	[-0.0869, -0.0649]	<0.001
MultiDiac	mT5 - ami	WER	-0.0266	[-0.0344, -0.0189]	<0.001
MultiDiac	mT5 - ami	CER	-0.0059	[-0.0098, -0.0018]	0.001
MultiDiac	mT5 - ami	DER	-0.0076	[-0.0293, +0.0143]	0.501
MultiDiac	mT5 - ami	WDER	-0.0190	[-0.0280, -0.0099]	<0.001
MultiDiac	mT5 - Unigram	WER	-0.0746	[-0.0856, -0.0644]	<0.001
MultiDiac	mT5 - Unigram	CER	-0.0225	[-0.0279, -0.0173]	<0.001
MultiDiac	mT5 - Unigram	DER	-0.0532	[-0.0790, -0.0276]	0.001
MultiDiac	mT5 - Unigram	WDER	-0.0515	[-0.0633, -0.0406]	<0.001
MultiDiac	ami - Unigram	WER	-0.0480	[-0.0591, -0.0375]	<0.001
MultiDiac	ami - Unigram	CER	-0.0166	[-0.0213, -0.0119]	<0.001
MultiDiac	ami - Unigram	DER	-0.0456	[-0.0680, -0.0223]	<0.001
MultiDiac	ami - Unigram	WDER	-0.0326	[-0.0438, -0.0214]	<0.001

D. Coverage Analysis

The coverage categories of both datasets are shown in Table 15, measured against the Menyo-20k training sentences. On exact token coverage (the first two categories) MultiDiac (88.59%) is close to WAEC (88.06%), and on surface token coverage it is higher (96.22% vs. 91.69%), so MultiDiac has fewer unseen words, but the performance of the models is worse on its WER, CER and WDER. This is the most informative result in the paper, and it is worth following carefully. These figures differ slightly from those of the earlier version of this study, which did not normalize Unicode before matching.

Table 15. Share of Reference Tokens and Types in Each Coverage Category Relative to Menyo-20k (%).

Category	WAEC tokens	MultiDiac tokens	WAEC types	MultiDiac types
Seen, majority form	69.82	59.52	49.49	40.50
Seen, minority form	18.24	29.08	17.76	27.82
Seen, other form	3.64	7.62	10.10	20.73
Unseen word	8.31	3.78	22.64	10.95
Total count	6,103	5,575	1,475	1,452

WAEC's difficulty is partly a coverage problem. It has 334 unseen word types, which account for 22.64% of its distinct diacritized types and 8.31% of its tokens. The most frequent of these are terms specific to the Yorùbá literary examination domain. The word akewi (praise poet) is found 14 times in the corpus, oriki (praise poetry) 11 times, and grammatical terms like konsonanti, faweli, isomoloruko are used several times in different places. All of these are not found in any form in the Menyo-20k training vocabulary. A model cannot recall the appropriate diacritization of a word that it has never seen, but it can try to use the phonological patterns that it has learned to make a guess. Table 16 shows that this works only in part: mT5 and ami are wrong on 64.50% and 64.10% of the unseen WAEC tokens, against 96.84% for Unigram, which leaves such words unchanged. The models therefore generalize beyond their vocabulary, but most unseen words are still wrong.

MultiDiac's difficulty is of a different nature. Its unseen word count is lower (159 types, 10.95% of its types and 3.78% of its tokens), and most of the frequent ones are personal names used as base words to construct ambiguous sentences: names like sola, solape, and yewande appear in specific tonal forms in the reference text but have limited representation in training. That is not the primary challenge, however. The more revealing figure is the share of tokens whose bare word was seen in Menyo-20k but whose required form is not the most frequent one: 36.70% of MultiDiac tokens (29.08% minority form and 7.62% other form) against 21.87% in WAEC (18.24% and 3.64%). This larger share reflects how MultiDiac was constructed: the base words were selected precisely because they appear in training data but are ambiguous.

Table 16. Error Rate by Coverage Category (Share of Reference Words not Reproduced Exactly).

Dataset	Category	Tokens	mT5	ami	byT5	omowe	Unigram
WAEC	Seen, majority form	4,261	0.0720	0.0519	0.3987	0.2072	0.0005
WAEC	Seen, minority form	1,113	0.2579	0.3720	0.6918	0.3603	1.0000
WAEC	Seen, other form	222	0.7883	0.8919	0.9459	0.8649	1.0000
WAEC	Unseen word	507	0.6450	0.6410	0.8955	0.7081	0.9684
MultiDiac	Seen, majority form	3,318	0.0741	0.0832	0.3487	0.2230	0.0000
MultiDiac	Seen, minority form	1,621	0.5879	0.6570	0.8291	0.6607	1.0000
MultiDiac	Seen, other form	425	0.9576	0.9671	0.9459	0.9459	1.0000
MultiDiac	Unseen word	211	0.8246	0.8246	0.8720	0.7915	0.6398

The categories show where the errors of the best models are. On words in their majority form, mT5 and ami are wrong on only 7.41% and 8.32% of the MultiDiac tokens (7.20% and 5.19% on WAEC). On words in a minority form they are wrong on 58.79% and 65.70% of the MultiDiac tokens, against 25.79% and 37.20% on WAEC, and on words seen only in another diacritized form the error rates are 78.83% to 96.71% on both datasets. Among the reference words that are not in the majority form, the models output the Menyo-20k majority form in 53.57% (mT5) and 58.31% (ami) of the cases on MultiDiac, compared with 22.92% and 31.84% on WAEC. The models thus do not follow the sentence context in every case but they often fall back on the most frequent form, and more so when the data is built around ambiguity. The harder challenge on MultiDiac is not unknown words but words the models have seen in one diacritized form when the context of the sentence calls for another.

This finding is related to the one made by Olawole et al. [12] that domain membership is a better predictor of performance than model architecture. It is also in line with the results of Orife et al. [11] that the best way for improving restoration quality is by increasing the domain coverage in the training data. The Menyo-20k corpus is varied by the standards of what is available for Yorùbá, as Table VI shows. But examination language, oral literary genres, and text constructed around lexically ambiguous target words are not among its sources. The coverage analysis puts this gap into perspective, although it is observational and does not by itself show that adding such data would remove the errors.

E. Model Size and Architecture

ami, with 1.29 million parameters and a BiLSTM architecture, is within a small margin of mT5 on both datasets. On WAEC, ami is better on CER (0.0807 vs. 0.1026, $p = 0.010$) and the two are tied on the other metrics. On MultiDiac, mT5 is ahead on WER, CER and WDER by small but reliable margins (0.0266, 0.0059 and 0.0190) and tied on DER. Both models used the Menyo-20k training split, but they were not trained on identical data: mT5 was also fine-tuned on JW300 and ami on the diacritized part of Yorùbá Text C3, and the two differ in architecture, tokenization and pretraining history. The comparison therefore cannot separate the effect of the training data from the effect of the architecture. What it shows is that a model more than 450 times larger does not give a large or

DOI: 10.34148/teknika.v15i3.1600

consistent advantage on these two datasets. This is not a claim that BiLSTMs are generally competitive with transformers. It is consistent with, but does not prove, the view that the distribution of the training text influences out-of-domain performance more than model size does. A controlled test would train both architectures on identical data, which is left for future work.

Ayoola and Olukanmi [13] noted a similar finding when they compared mT5 with frontier LLMs with a significantly greater number of parameters on YAD. It was the fine-tuning of the Yorùbá, not scale, that was important. Gessler and Zeldes [29] noted more broadly for low-resource NLP that the gap between simple and complex models shrinks when training data is limited, because additional parameters require additional data to be useful. The present results are in line with both observations on out-of-domain text, but they do not test them directly.

F. Comparison with Toyin et al.

Toyin et al. [7] evaluated omowe-t5 and byT5, among other specialized models and LLMs, on the Yorùbá test split of MultiDiac and reported CERs of 52.84% for omowe-t5 and 46.50% for byT5, so byT5 was the better of the two there. These figures cannot be compared directly with the present ones, because their test split of 101 samples is not in the public release and the pipeline and decoding settings differ. On the public samples, the CER order of the two models depends on the subset: on the 603 samples byT5 has the lower CER (0.3223 vs. 0.3826), while on the 41 dev samples omowe has the lower CER (0.2711 vs. 0.3056). On DER and WDER, omowe is clearly better than byT5 in both subsets (full set DER 0.6778 vs. 0.8596 and WDER 0.5772 vs. 0.8219). The order of these two models therefore depends on the metric as well as on the subset.

The more important methodological point is that specialized T5 models can produce outputs that differ from the input text. Toyin et al. report hallucination WERs of 49.67 for omowe-t5 and 35.41 for byT5 in the arXiv version of their paper, and such outputs inflate every error metric, because missing words are counted as deletions and surplus text as insertions. When these figures are compared against LLM outputs that are complete, the comparison is not measuring the same thing. The conclusion from that paper that LLMs outperform specialized ADR models for Yorùbá should be interpreted with this in mind. The mT5 model in that comparison was a different checkpoint (Davlan/mt5-small-diacritizer-menyo in the arXiv version) from the mT5_base_yoruba_adr evaluated here, and mT5_base_yoruba_adr, the strongest model in this study by most metrics, was not part of it, so the ranking of specialized models in [7] does not carry over to it.

G. Strengths and Limitations

The study covers two out-of-domain sets of different character: examination text with many unseen domain terms, and sentences built around ambiguous words. It adds two baselines, reports paired bootstrap intervals and tests, separates unseen words from seen words in a less frequent form, and records the data statistics, checkpoint revisions and software environment so that the evaluation can be repeated.

Several limitations apply. The datasets are small (664 and 603 samples, and 41 for the dev split), which leaves DER intervals about 0.05 wide on each side. The WAEC references were corrected by the authors without a second annotator, and some may still lack marks that the models restore, for example fófó against fọ́fọ́ in Table IX, which would be counted as errors. The independent check of MultiDiac is described for its test split only, and its references do not mark the syllabic nasal consistently in the samples inspected. Menyo-20k is used as a proxy for training exposure, but three of the four models were also trained on other corpora, so some words counted as unseen may have been seen. The design is observational, with no model trained under controlled conditions. DER and WDER compare positions by index, which penalizes outputs of a different length, and scoring was done without Unicode normalization on strings that include 19 WAEC and 13 MultiDiac samples with mixed forms. The category analysis depends on the word alignment and on majority forms taken from a single corpus. Each model was run once, and the changed scores of byT5 and omowe in the rerun show that results for some checkpoints depend on the runtime environment.

V. CONCLUSION

This paper evaluated four ADR models on two out-of-distribution datasets, the WAEC examination corpus and MultiDiac, to understand what drives performance when models encounter text unlike their training data. The central finding is that vocabulary coverage alone does not explain model difficulty. MultiDiac has fewer unseen words than WAEC (3.78% vs. 8.31% of tokens) but more words that were seen in Menyo-20k only in another diacritized form (36.70% vs. 21.87%), and the word-level scores of all models are worse on it. The harder challenge is not unknown words but known words that appear in training under one diacritized form when the correct form, given the sentence context, is another. On words in a minority form, mT5 and ami are wrong 58.79% and 65.70% of the time on MultiDiac, against 7.41% and 8.32% on words in their majority form. A 1.29M-parameter BiLSTM performs close to a 582M-parameter transformer: the two do not differ significantly in DER on either dataset and differ by at most 0.03 on the other metrics. The models differ in architecture and training data, so this is consistent with, but does not show, that the training corpus matters more than the architecture. A context-free lookup of the most frequent form from Menyo-20k is within 0.12 WER of the best model, and byT5 and omowe score worse than it on both datasets.

These results come with limitations. The WAEC corpus covers eight examination years. This is not the exhaustive list. A wider coverage could introduce a different register of literary Yorùbá. The WAEC references were corrected by the authors without a second annotator. The coverage analysis treats Menyo-20k as a proxy for all training exposure, but three of the four models were additionally trained on other corpora,

meaning some terms counted as unseen may have been seen during training. The MultiDiac evaluation uses the combined train and dev splits because the original test split is not public, which limits direct comparability with the figures of the original paper, and the dev split is reported separately. The datasets are small, DER and WDER are computed by index, and the scores of byT5 and omowe changed between two runs.

The results suggest that the next step for Yorùbá ADR is to have more data before it requires more model. The current training corpus has taken the field to a helpful, but limited, baseline. What is needed is diacritized Yorùbá text from domains that are not currently represented, such as examination papers, literary criticism, oral tradition, and the types of everyday Yorùbá writing that are created in professional and institutional settings. The architecture can follow once the data is there to learn from. Future work should train both small and large architectures on identical data to test this directly, add a second annotator for the WAEC references, and score diacritics after alignment instead of by index.

REFERENCES

[1] I. Shode, D. I. Adelani, and A. Feldman, "YOSM: A new Yorùbá sentiment corpus for movie reviews," in Proc. ICLR 2022 Workshop AfricaNLP, 2022. [Online]. Available: https://openreview.net/forum?id=rRzx5qzVIb9

[2] A. Aremu, J. O. Alabi, and D. I. Adelani, "YORC: Yorùbá Reading Comprehension Dataset," in Proc. Seventh Widening NLP Workshop (WiNLP 2023), 2023.

[3] T. V. Asubiaro, "Statistical patterns of diacritized and undiacritized Yorùbá texts," Journal of Information Science Theory and Practice, 2015.

[4] J. C. Wells, "Orthographic diacritics and multilingual computing," Language Problems and Language Planning, vol. 24, no. 3, pp. 249-272, 2000, doi: 10.1075/lplp.24.3.04wel.

[5] D. I. Adelani, D. Ruiter, J. O. Alabi, D. Adebonojo, A. Ayeni, M. Adeyemi, A. E. Awokoya, and C. España-Bonet, "The effect of domain and diacritics in Yorùbá-English neural machine translation," in Proc. Machine Translation Summit XVIII: Research Track, 2021, pp. 61-75. [Online]. Available: https://aclanthology.org/2021.mtsummit-research.6

[6] kenny0bi, "ami-yoruba-diacritics," Hugging Face, 2023. [Online]. Available: https://huggingface.co/kenny0bi/ami-yoruba-diacritics

[7] H. O. Toyin, S. M. Magdy, and H. Aldarmaki, "Are LLMs good text diacritizers? An Arabic and Yorùbá case study," in Proc. LREC 2026, 2026, pp. 580-589.

[8] A. Akinlabi and T. Sanghal, "Improving English-Yorùbá neural machine translation through dataset curation, diacritic restoration, and fine-tuning of mT5 models," UC Berkeley Recent Work, 2025. [Online]. Available: https://escholarship.org/uc/item/1rw396k3

[9] T. P. Adewumi, F. Liwicki, and M. Liwicki, "The challenge of diacritics in Yorùbá embeddings," in ML4D Workshop at NeurIPS 2020, 2020. [Online]. Available: https://arxiv.org/abs/2011.07605

[10] I. Orife, "Attentive sequence-to-sequence learning for diacritic restoration of Yorùbá language text," in Proc. Interspeech 2018, 2018, pp. 2848-2852, doi: 10.21437/Interspeech.2018-42.

[11] I. Orife, D. I. Adelani, T. Fasubaa, V. Williamson, W. F. Oyewusi, O. Wahab, and K. Tubosun, "Improving Yorùbá diacritic restoration," in Proc. ICLR 2020 AfricaNLP Workshop, 2020. [Online]. Available: https://arxiv.org/abs/2003.10564

[12] A. M. Olawole, J. O. Alabi, A. K. Sakpere, and D. I. Adelani, "YAD: Leveraging T5 for improved automatic diacritization of Yorùbá text," in Proc. AfricaNLP Workshop at ICLR 2024, 2024. [Online]. Available: https://arxiv.org/abs/2412.20218

[13] J. I. Ayoola and P. O. Olukanmi, "Automatic diacritization models for a high-population low-resource African language (Yorùbá)," Applied Sciences, vol. 16, no. 12, p. 6195, 2026, doi: 10.3390/app16126195.

[14] A. Adegoke-Elijah, O. A. Odejobi, and A. S. Salawu, "Lexical ambiguity resolution system for standard Yorùbá verbs," American Journal of Engineering Research, vol. 7, no. 6, pp. 170-176, 2018.

[15] A. A. Akindele, "Resolving lexical verb ambiguity in Yorùbá through semantic web annotations," Yoruba Studies Review, vol. 8, no. 1, 2023, doi: 10.32473/ysr.8.1.134096.

[16] I. Adebara and M. Abdul-Mageed, "Towards Afrocentric NLP for African languages: Where we are and where we can go," in Proc. 60th Annual Meeting of the Association for Computational Linguistics, 2022, pp. 3300-3319, doi: 10.18653/v1/2022.acl-long.265.

[17] I. Inuwa-Dutse, "NaijaNLP: A survey of Nigerian low-resource languages," arXiv preprint arXiv:2502.19784, 2025. [Online]. Available: https://arxiv.org/abs/2502.19784

[18] T. A. Jimoh, T. De Wille, and N. S. Nikolov, "Bridging gaps in natural language processing for Yorùbá: A systematic review of a decade of progress and prospects," Natural Language Processing Journal, vol. 13, p. 100194, 2025, doi: 10.1016/j.nlp.2025.100194.

[19] Lexilogos, "Yoruba keyboard," Lexilogos. [Online]. Available: https://www.lexilogos.com/keyboard/yoruba.htm

[20] Davlan, "mT5_base_yoruba_adr," Hugging Face. [Online]. Available: https://huggingface.co/Davlan/mT5_base_yoruba_adr

[21] Davlan, "byt5-small-diacritizer-menyo," Hugging Face. [Online]. Available: https://huggingface.co/Davlan/byt5-small-diacritizer-menyo

[22] Davlan, "omowe-t5-small-diacritizer-all-und-full," Hugging Face. [Online]. Available: https://huggingface.co/Davlan/omowe-t5-small-diacritizer-all-und-full

CC BY SA

[23] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "ByT5: Towards a token-free future with pre-trained byte-to-byte models," Transactions of the Association for Computational Linguistics, vol. 10, pp. 291-306, 2022, doi: 10.1162/tacl_a_00461.

[24] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38-45, doi: 10.18653/v1/2020.emnlp-demos.6.

[25] V. I. Levenshtein, "Binary codes capable of correctingdeletions, insertions and reversals," Soviet Physics Doklady, vol. 10, no. 8, pp. 707-710, 1966.

[26] T. von Neumann, C. Boeddeker, K. Kinoshita, M. Delcroix, and R. Haeb-Umbach, "On word error rate definitions and their efficient computation for multi-speaker speech recognition systems," in Proc. ICASSP 2023, Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10094784.

[27] T. D. K, J. James, D. P. Gopinath, and M. A. K, "Advocating character error rate for multilingual ASR evaluation," in Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, Apr. 2025, pp. 4941-4950, doi: 10.18653/v1/2025.findings-naacl.277.

[28] I. Zitouni, J. S. Sorensen, and R. Sarikaya, "Maximum entropy based restoration of Arabic diacritics," in Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 2006, pp. 577-584, doi: 10.3115/1220175.1220248.

[29] L. Gessler and A. Zeldes, "MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning," in Proc. 2nd Workshop on Multi-lingual Representation Learning (MRL), Abu Dhabi, United Arab Emirates, 2022, pp. 86-99.

[30] P. Koehn, "Statistical significance tests for machine translation evaluation," in Proc. 2004 Conf. Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp. 388-395.

[31] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap. New York, NY, USA: Chapman & Hall, 1993.