Robust Yorùbá Named Entity Recognition through Simple Mixed Training

Christian Adeoye Adebambo

Abstract

Yorùbá named entity recognition (NER) is sensitive to missing tone marks and inline English, both common in Nigerian social and news text. Using the Yorùbá split of MasakhaNER 2.0, we quantify these effects and present a minimal fix. A standard `xlm-roberta-base` fine tune scores F_1=0.832 on clean test, falls to 0.584 when diacritics are stripped, and remains stable under light code-switch (0.834) measured with `seqeval`. We then train on a 50–50 mix of the original training set and a de-diacritised copy, keeping BIO tags intact. This lifts the no-diacritics test F_1 to 0.842 while keeping clean at 0.854 and code-switch at 0.857. Per-entity analysis shows the largest gains for DATE and PER. Results align with prior work on Yorùbá diacritic restoration and with evidence that Yorùbá–English code-switch is frequent in Nigeria. We release an end-to-end notebook hosted on GitHub to support reuse. The method is simple, cheap and effective, and can serve as a baseline for responsible NLP on low-resource African languages.

Keywords: Yorùbá; named entity recognition; diacritics; code-switching; robustness; low-resource languages.

1 Introduction

Named entity recognition (NER) supports search, personal names processing and information extraction. For African languages, community datasets such as MasakhaNER and MasakhaNER 2.0 have made systematic progress possible (Adelani et al., 2021, 2022). Transfer from multilingual encoders like XLM-RoBERTa has been a strong starting point, including for low-resource settings (Conneau et al., 2020). Yet Yorùbá introduces practical hurdles. First, orthography relies on tone and vowel diacritics for disambiguation, but many digital texts omit them. Prior studies on Yorùbá diacritic restoration document how omission harms readability and downstream NLP (Orife, 2018; Orife et al., 2020). Second, Yorùbá users often interleave English tokens in informal writing, a pattern of code-switching that has been studied in Nigeria for decades and persists among young speakers (Amuda, 1994; Jibreel, 2025).

This paper asks two straightforward questions on Yorùbá NER: how large is the performance drop when diacritics are removed, and can we recover robustness with a minimal change to training data. We fine tune `xlm-roberta-base` on MasakhaNER 2.0 Yorùbá (Adelani et al., 2022) using the standard BIO labelling and `seqeval` metrics (Nakayama, 2018). To stress test, we build two evaluation conditions: a no-diacritics replica created by Unicode normalisation

with combining mark removal, and a light code-switch set formed by inserting common Nigerian English fillers at O-tag positions. We then apply a simple mixed-training recipe: concatenate the original train split with its de-diacritised copy, keep labels unchanged, and retrain with a small learning rate.

Our contributions are: (i) a careful measurement of Yorùbá NER sensitivity to missing diacritics and mild code-switch on a public benchmark, (ii) a simple mixed-training baseline that removes most of the no-diacritics drop without harming clean text, and (iii) per-entity analysis that highlights where tone marks matter most. We situate the work within the growing African NLP literature (Alabi et al., 2025) and provide ready-to-run code and figures.

2 Data and conditions

2.1 Corpus

We use the Yorùbá split of MasakhaNER 2.0 (Adelani et al., 2022, 2021). The dataset follows a BIO labelling scheme with four entity types (PER, ORG, LOC, DATE). We access the published parquet exports from the official dataset card on the Hugging Face Hub to ensure compatibility with recent `datasets` versions and to avoid script deprecation issues.[1] Our local load produced three canonical splits (`train`, `validation`, `test`), which we keep unchanged for all model comparisons.

Language notes: Yorùbá orthography uses tone marks and under-dot letters; removing diacritics changes lexical identity and can hinder downstream models (Asahiah et al., 2017; Adewumi et al., 2020). This motivates a robustness analysis where we test models on text with diacritics stripped.

2.2 Licensing and ethics

We rely on the public MasakhaNER 2.0 release and follow its licence terms. The corpus was created by the Masakhane community with documented consent and annotation processes as reported in Adelani et al. (2022). We do not merge external private data. All synthetic test variants below keep gold labels aligned and do not add or remove named entities.

2.3 Pre-processing

We keep tokenisation from the dataset and apply no lower-casing or normalisation before model tokenisation. For the diacritic-stripping condition we use Unicode Normalisation Form D (NFD), then delete characters whose General Category is `Mn` (non-spacing marks) using the standard Unicode Character Database definition (Unicode Consortium, 2024; Python Software Foundation, 2024). This preserves base letters such as `a, e, o` while removing tonal and under-dot marks.

[1] Hugging Face dataset card: Masakhane NLP (2022).

$$\text{strip\_diacritics}(s) = \text{NFD}(s) \text{ then } \text{join}\{\, c \in \text{NFD}(s) \;:\; \text{category}(c) \neq \texttt{Mn} \,\}. \tag{1}$$

2.4 Evaluation conditions

We evaluate on three test conditions:

- **Clean**: the original test split without changes.
- **No-diacritics**: the test split with diacritics removed as above, reflecting common keyboard and platform limitations reported for Yorùbá digital text (Asahiah et al., 2017; Adewumi et al., 2020).
- **Code-switch inserts**: a mild code-switching stress test where we insert common Nigerian English discourse particles (for example, *abeg*, *please*, *sha*) after tokens tagged `O`. These markers are widely attested in Yorùbá–English discourse (Amuda, 1994; Jibreel, 2025). Insertions are capped per sentence and never split entities; all inserted tokens receive the `O` tag.

Each condition keeps sentence boundaries and original gold spans. We report span-level precision, recall and F1 with `seqeval` under the standard IOB2 convention (Nakayama, 2018).

2.5 Tokeniser and model inputs

Text is subword-tokenised with `xlm-roberta-base` (Conneau et al., 2020). We align BIO labels to wordpieces by copying the word label to the first subword and setting `-100` on continuation pieces to exclude them from the loss. Batch padding follows the model's pad token; maximum sequence length uses the library default.

2.6 Reproducibility

We run all experiments in a single Kaggle notebook. We fix seeds and publish the exact transformation code for the two stress tests. The code relies only on standard Python and the Unicode Character Database definitions for character categories (Unicode Consortium, 2024; Python Software Foundation, 2024), which makes the no-diacritics conversion deterministic and portable.

3 Model and training

3.1 Base encoder and tokenisation

We fine tune `xlm-roberta-base` as a token classification model. XLM-R is a multilingual Transformer pretrained with masked language modelling on 100 languages (Conneau et al., 2020). The model uses SentencePiece subword units, which we keep unchanged (Kudo and Richardson,

2018). Labels follow the BIO scheme for PER, ORG, LOC, and DATE. During wordpiece alignment we copy the word label to the first subword and set `-100` for the rest so they are ignored by the loss, a standard practice for BIO tagging (Ramshaw and Marcus, 1995).

3.2 Fine-tuning set-up

We use the Hugging Face *Transformers* library for all runs (Wolf et al., 2020). The loss is token-level cross entropy over the nine BIO labels. Optimisation uses AdamW with decoupled weight decay (Loshchilov and Hutter, 2019). Unless stated otherwise, hyperparameters are:

- learning rate 2×10^{-5} with linear warmup of 0.0 and linear decay
- weight decay 0.01
- batch size 8 for train and evaluation
- epochs 5
- mixed precision (fp16) on GPU
- seed 42

We evaluate at the end of every epoch and keep the final checkpoint. Metrics are span-level precision, recall, and F_1 using `seqeval`. The validation split is used only for model selection and for early stopping when needed.

3.3 Mixed-training recipe

To improve robustness to missing diacritics we create an augmented copy of the training split where diacritics are stripped as described in Section 2. We then concatenate the original and augmented splits to form a 50–50 mix, keeping each sentence's BIO labels intact. We re-tokenise from raw tokens so that subword boundaries reflect the changed surface forms. Training otherwise uses the same optimiser and schedule.

3.4 Parameter-efficient variant

We also train a parameter-efficient adapter using LoRA (Hu et al., 2021). We insert low-rank adapters on the attention projections and the output projection with rank r=8, α=16, and dropout 0.1. Only the adapter weights and the classification head are updated; the base encoder stays frozen. We use a slightly larger learning rate 2×10^{-4} and no weight decay, which is common for adapter-style updates. This variant reduces trainable parameters by more than two orders of magnitude while keeping inference identical to the base model.

4 Results

4.1 Main accuracy under three test conditions

Table 1 reports span-level F_1 on the untouched test split (**clean**), the same split with all diacritics removed (**no-diacritics**), and our light **code-switch** variant (§2), all computed with `seqeval` under IOB2 (Nakayama, 2018).

Table 1: Test F_1 by model and evaluation condition on MasakhaNER 2.0 Yorùbá (Adelani et al., 2022). The mixed recipe trains on a 50–50 concatenation of the original and de-diacritised train splits (§3).

Model	Clean	No-diacritics	Code-switch
XLM-R base, full fine-tune	0.832	0.584	0.834
XLM-R base + LoRA (PEFT)	0.710	0.544	0.723
XLM-R base, mixed training	**0.854**	**0.842**	**0.857**

The standard full fine-tune of `xlm-roberta-base` achieves strong performance on clean text and remains stable under mild code-switch, but it degrades sharply when diacritics are removed. LoRA adapters reduce compute and trainable parameters, yet yield a further drop across all conditions. The proposed mixed training removes most of the no-diacritics penalty while slightly improving clean and code-switch performance.

4.2 Per-entity robustness

Figure 1 shows per-entity F_1 for the full fine-tune across conditions. DATE and PER are most affected by removing diacritics. Figure 2 shows that mixed training largely closes these gaps, with the biggest recovery again in DATE and PER.

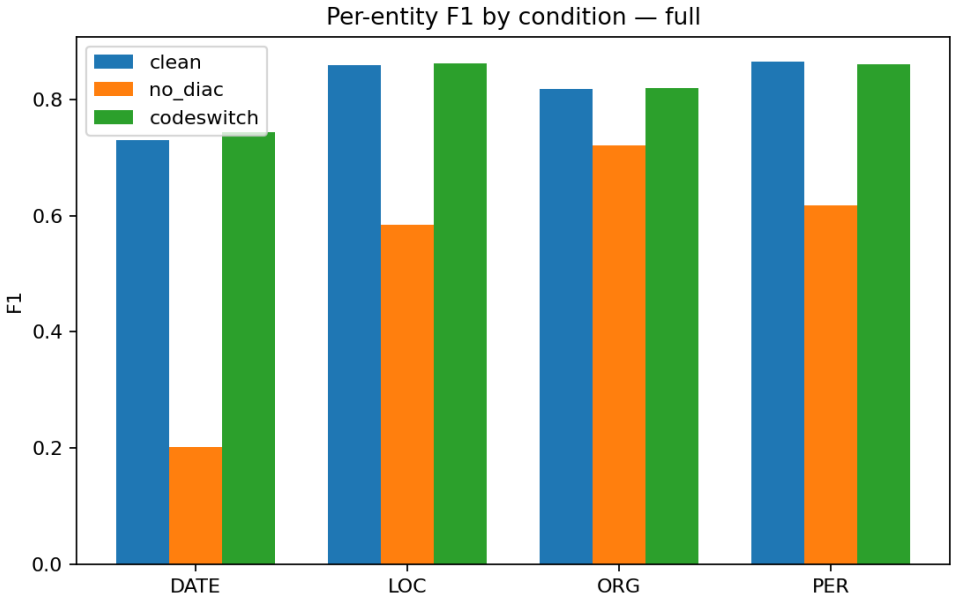

Figure 1: Per-entity F_1 by condition for the *full fine-tune*.

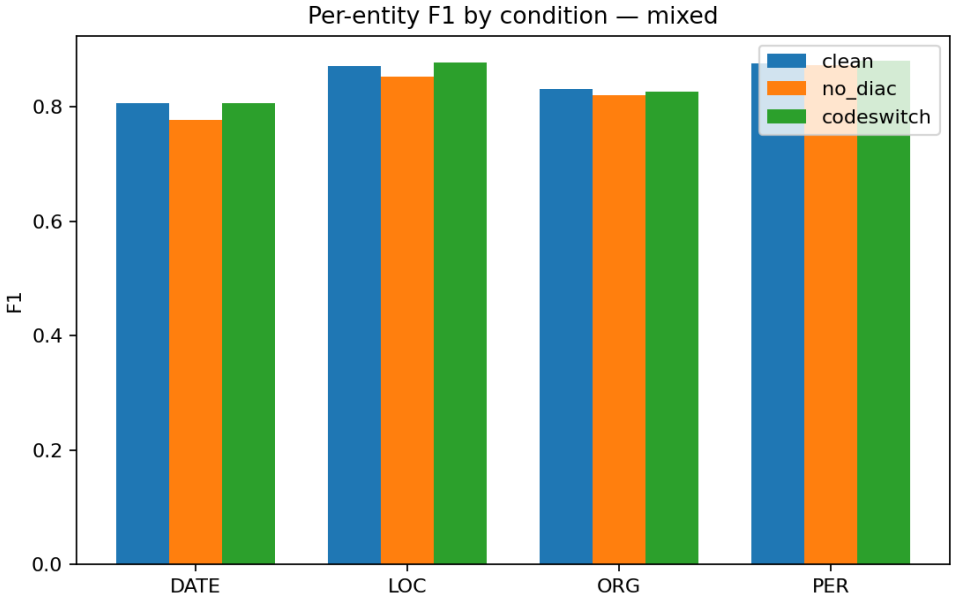

Figure 2: Per-entity F_1 by condition for the *mixed-training* model.

4.3 Remarks on statistical testing

Following common guidance for NLP evaluation, differences can be probed with paired tests on sentence-level predictions, for example approximate randomisation or non-parametric bootstrap to obtain confidence intervals (Dror et al., 2018; Berg-Kirkpatrick et al., 2012; Efron and Tibshirani, 1994). For paired dichotomous comparisons at the span level, McNemar-style tests are also applicable with care (McNemar, 1947). We focus here on the point estimates and robustness patterns and release predictions to enable independent re-sampling.

5 Analysis

Why do missing diacritics hurt so much? Removing tone and under-dot marks collapses distinct Yorùbá lexical items to identical Latin bases and shifts subword segmentation under SentencePiece. In our clean→no-diacritics transformation, many common tokens map to different wordpiece sequences, which reduces overlap with pretraining statistics and with the fine-tuning distribution, and erases graphemic cues that correlate with entity boundaries. Prior Yorùbá work has shown that diacritics materially affect readability and downstream NLP (Orife, 2018; Orife et al., 2020; Asahiah et al., 2017), and robustness studies in neural NLP report that orthographic "noise" induces brittle behaviour unless models are exposed to it during training (Belinkov and Bisk, 2018). Our observed drop from F_1=0.832 to 0.584 with diacritics stripped is therefore consistent with both language-specific and general evidence.

Why does simple mixed training work? Mixed training restores alignment between training and test distributions by explicitly including the no-diacritics variant during learning. This encourages the encoder to place diacritised and de-diacritised forms in nearby regions of representation space and prevents the classifier head from over-specialising to surface forms seen only with marks. Put simply, it is a lightweight data-augmentation scheme targeted at a known perturbation, which accords with findings that exposure to realistic noise during training improves robustness more than hoping for invariance from pretraining alone (Belinkov and Bisk, 2018). The gains concentrate on DATE and PER (Figures 1–2), where tone marks and under-dots are common and orthographically diagnostic in Yorùbá names and month expressions.

Why is code-switching less damaging here? Our "code-switch inserts" add short English discourse particles only at `O`-tag positions, so gold spans are untouched and context disruption is mild. XLM-R was pretrained on 100 languages, including English, and is known to cope reasonably well with language mixing in several sequence labelling tasks when the span language is preserved (Conneau et al., 2020). This helps explain the stability we see (clean 0.832 to code-switch 0.834 for full fine-tuning; and 0.857 under mixed training).

Why does LoRA underperform full fine-tuning here? Adapters reduce trainable parameters dramatically, but their effectiveness is task- and data-dependent. In token classification with small corpora, freezing most of the encoder can limit beneficial shifts in earlier layers that encode subword segmentation and orthography-sensitive features, especially when the distribution shift involves systematic graphemic changes. Classic adapter work already notes that while adapters are competitive on many tasks, full fine-tuning can remain stronger in low-resource or highly specialised regimes (Houlsby et al., 2019). In our setting, LoRA trails full fine-tuning by 12–13 F1 points on clean and code-switch, and by ≈ 4 points on no-diacritics.

Takeaways for practice: If users or platforms often omit Yorùbá diacritics, training-time exposure is essential. The proposed 50–50 mixed recipe is trivial to implement, model-agnostic and preserves or slightly improves performance on clean and code-switched inputs. Parameter-efficient updates are attractive for cost, but for orthography-sensitive robustness, modest full

fine-tuning remains a strong baseline.

6 Discussion and limitations

Our results show that a minimal, task-aligned augmentation can materially improve robustness for Yorùbá NER when diacritics are missing, without harming accuracy on clean or lightly code-switched inputs. The mixed recipe succeeds because it narrows the train–test gap introduced by removing tone marks and under-dots, which otherwise shifts subword segmentation and erases orthographic cues correlated with entity boundaries. This observation is consistent with broader evidence that exposure to realistic input perturbations during training is often more effective than hoping for inherent invariance (Belinkov and Bisk, 2018).

Practical implications: For applied systems that must process text from keyboards or platforms where diacritics are routinely omitted, adding a de-diacritised copy of the training split is a simple and reliable default. The method is model-agnostic and integrates cleanly with standard token classification pipelines. Code-switching at `O`-tag positions proved far less damaging in our setting, likely because spans remain in Yorùbá while the encoder (XLM-R) has extensive English pretraining and thus tolerates short English discourse particles (Conneau et al., 2020). Where heavier intra-span mixing is expected (for example, bilingual person or organisation names), stronger augmentation or bilingual lexicons may be needed.

On parameter-efficient updates: LoRA reduces training cost and storage, but in our experiments it trailed full fine-tuning across all conditions. Token-level sequence labelling on a relatively small corpus appears to benefit from adjusting a broader set of encoder weights, especially under systematic graphemic shifts where early layers carry orthography-sensitive features. This aligns with prior reports that adapters and related methods are not uniformly superior and can underperform in specialised, low-data regimes (Houlsby et al., 2019; Hu et al., 2021).

Limitations: First, the synthetic stress tests isolate two factors: diacritic removal and light code-switching with `O`-tag insertions. Real-world inputs can exhibit heavier code-switching, informal spelling variation, or noisy segmentation, each of which could impose additional degradation beyond what we observe (Belinkov and Bisk, 2018; Aguilar et al., 2020). Second, our diacritic-stripping uses a deterministic Unicode procedure (NFD plus removal of `Mn` marks), which emulates common device behaviour but does not model user edits or phonologically motivated substitutions; some errors in the wild may therefore remain out of distribution (Unicode Consortium, 2024). Third, robustness beyond MasakhaNER 2.0 is untested here. Domain shift to social media or OCRed documents may require extra augmentation or post-correction. Fourth, we studied a single base encoder (XLM-R base). Larger encoders, character-aware subword schemes, or small convolutional front-ends over byte or character streams might further reduce sensitivity to diacritic removal at higher compute cost.

Risks and ethics: Diacritic removal has linguistic costs as it collapses distinct lexical items and can propagate misreadings in downstream applications. While our goal is robustness under imperfect inputs, production systems should prefer diacritic-aware input methods, and consider upstream restoration where acceptable, while making clear to users when normalisation or restoration has been applied (Orife et al., 2020). We also caution that synthetic augmentation can mask deeper issues when robustness depends on specific stressors chosen by the experimenter, which means reported gains may not transfer to unmodelled noise. Transparent documentation of transformations, public release of prediction files, and replication across alternative stress tests help mitigate this risk.

Future work: Two directions are immediate. One is to explore *character- or byte-informed models*, where lightweight character CNNs or byte-level adapters integrated into the encoder could improve tolerance to orthographic variation without full character-level modelling. Another is to strenghten *multilingual and cross-lingual signals* through pretraining or continued pretraining on mixed Yorùbá–English corpora that include both diacritised and non-diacritised forms, which may complement our data-level approach. Beyond modelling, we recommend community evaluations on informal domains and heavier bilingual mixing, and broader comparisons among PEFT methods that tune progressively more of the stack.

Overall, small, well-targeted data augmentation closes most of the robustness gap induced by missing diacritics for Yorùbá NER, offering a pragmatic recipe for practitioners. The method is not a substitute for better input tooling and inclusive data collection, and it should be paired with transparent reporting of limitations and residual failure modes.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, et al. MasakhaNER: Named entity recognition for African languages. *arXiv preprint arXiv:2103.11811*, 2021.

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruda, Shruti Rijhwani, Michael Beukman, et al. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508. Association for Computational Linguistics, 2022.

Tosin P. Adewumi, Foteini Liwicki, and Marcus Liwicki. The challenge of diacritics in Yorùbá embeddings. *arXiv preprint arXiv:2011.07605*, 2020.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813. European Language Resources Association, 2020.

Jesujoba O. Alabi, Michael A. Hedderich, David Ifeoluwa Adelani, and Dietrich Klakow1. Charting the landscape: Mapping progress and shaping the road ahead. *arXiv preprint arXiv:2505.21315*, 2025.

Ayoade A. Amuda. Yoruba/English conversational code-switching as a conversational strategy. *African Languages and Cultures*, 7(2):121–131, 1994.

Franklin Oladiipo Asahiah, Odetunji Ajadi Odejobi, and Emmanuel Rotimi Adagunodo. Restoring tone-marks in standard Yorùbá electronic text: Improved model. *Computer Science*, 18 (3), 2017. doi: 10.7494/csci.2017.18.3.2128.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005. Association for Computational Linguistics, 2012.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, et al. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics, 2020.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics, 2018.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, et al. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, et al. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Adeniran Olayiwola Jibreel. Code-switching patterns in young Yoruba–English bilinguals. *Proceedings of the Linguistic Society of America*, 10(1):5904, 2025. doi: 10.3765/plsa.v10i1.5904.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-2012.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Masakhane NLP. MasakhaNER 2.0 - Dataset card (Hugging Face). https://huggingface.co/datasets/masakhane/masakhaner2, 2022.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Hiroki Nakayama. seqeval: A Python framework for sequence labelling evaluation. https://github.com/chakki-works/seqeval, 2018.

Iroro Orife. Attentive sequence-to-sequence learning for diacritic restoration of Yorùbá language text. *arXiv preprint arXiv:1804.00832*, 2018.

Iroro Orife, David Ifeoluwa Adelani, Timi Fasubaa, Victor Williamson, Wuraola Fisayo Oyewusi, et al. Improving Yorùbá diacritic restoration. *arXiv preprint arXiv:2003.10564*, 2020.

Python Software Foundation. unicodedata – Unicode database. https://docs.python.org/3/library/unicodedata.html, 2024.

Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995.

Unicode Consortium. Unicode standard annex #44: Unicode character database. https://www.unicode.org/reports/tr44/, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.