



Research Article

Bridging the NLP Gap: A Transformer-Based Sentiment Analysis Framework for Code-Mixed Nigerian Pidgin

Adewuyi Joseph Oluwaseyi¹ , Odiase Sophia Osariemen^{2*} , Imhanzenobe Daisy Ehijie³  and Ibrahim Mujeib Olayemi⁴ 

^{1,2,3,4}Department of Computer Science, School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria

Article Information

Abstract

Article History

Received: 27 May 2026

Revised: 30 June 2026

Accepted: 2 August 2026

Published online: 16 September 2026

Keywords

Nigerian Pidgin

Sentiment Analysis

Natural Language Processing

AfriBERTa

AfroXLM-R

Transformer Models

Low-Resource Languages

Deep Learning

Correspondence*

sophiacyber001@gmail.com

ORCID

Adewuyi Joseph Oluwaseyi 
<https://orcid.org/0009-0000-2325-7325>

Odiase Sophia Osariemen 
<https://orcid.org/0009-0007-9707-7651>

Imhanzenobe Daisy Ehijie 
<https://orcid.org/0009-0009-0950-3142>

Ibrahim Mujeib Olayemi 
<https://orcid.org/0009-0004-2560-9530>

This study addresses the limitations of existing Natural Language Processing (NLP) systems in analyzing Nigerian Pidgin, a widely spoken but low-resource language. Traditional sentiment analysis models, primarily designed for English, perform poorly due to challenges such as non-standardized spelling, semantic drift, and code-mixing. These limitations hinder accurate interpretation of public opinion expressed in Nigerian Pidgin. This project therefore aims to develop an effective sentiment analysis system capable of classifying Pidgin text into positive, negative, and neutral categories. A transformer-based approach was adopted using pretrained multilingual models, specifically AfriBERTa and XLM-R. The system was trained on the Nigerian Pidgin subset of the AfriSenti dataset (approximately 14,000 labeled tweets). A preprocessing pipeline involving data cleaning, normalization of spelling variations, and SentencePiece tokenization was implemented to improve model performance. The models were fine-tuned using supervised learning and evaluated using accuracy, precision, recall, and weighted F1-score. The model achieved a peak accuracy of 70.25%, with a precision of 0.6805, recall of 0.7025, and an F1-score of 0.6819. Comparative evaluation showed AfroXLM-R performed slightly better with 71.10% accuracy, emphasizing the advantage of African language-specific pretraining. These results demonstrate that transformer models can effectively capture sentiment in Nigerian Pidgin despite its linguistic complexity. In conclusion, the study confirms the viability of transformer-based NLP for low-resource languages. While challenges such as sarcasm, code-mixing, and limited datasets remain, the system provides a strong foundation for real-world applications and future improvements in indigenous language processing.

© 2026 Centre for Research and Innovation (CRI). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

I. INTRODUCTION

Natural Language Processing (NLP) research disproportionately favors high-resource languages like English and Mandarin. Consequently, widely spoken African languages remain structurally marginalized in artificial intelligence development. Nigerian Pidgin, an English-based creole spoken by an estimated 75 to 100 million people across West Africa, exemplifies this disparity. While it functions as a primary lingua franca for millions, it is categorized as a low-resource language in NLP. This classification stems from a severe lack of annotated datasets and the absence of a

standardized orthography [1]. Sentiment analysis algorithms typically struggle with these exact variables. We developed this study to build a functional sentiment analysis framework specifically adapted for Nigerian Pidgin. Using pretrained transformer models, our system classifies Pidgin text into positive, negative, or neutral categories. Doing so allows us to process regional public opinion and behavioral trends without forcing the data through an English-centric linguistic filter. Deep learning and transformer models have rapidly advanced NLP, but these breakthroughs mostly benefit high-resource languages like English, Chinese, and Spanish. African languages are consistently left behind. Nigerian

Pidgin is a perfect example: millions speak it daily across Nigeria, dominating informal conversations and social media, yet it receives very little computational attention.

The language itself makes traditional sentiment analysis difficult. It lacks standardized spelling, relies heavily on informal grammar, and constantly code-mixes with English. A sentiment classifier trained on standard English breaks down when faced with Pidgin's unique idioms and orthographic variations. Existing NLP architectures frequently fail when applied to Nigerian Pidgin. Standard sentiment analysis models show significant technical limitations across several specific areas. Firstly, standard English sentiment models like VADER or TextBlob rely on fixed lexicons with static polarities. Nigerian Pidgin frequently exhibits semantic drift, where words invert their traditional English meanings. For instance, the word "bad" often carries a positive sentiment in Pidgin, as in "Confirm, that food too bad!" Standard models cannot interpret these contextual polarity shifts.

Secondly, the absence of standardized spelling creates high out-of-vocabulary rates. A model trained to recognize "wahala" will usually fail to process "wahalla," treating them as entirely unrelated tokens. This orthographic inconsistency fragments the data density, which prevents the model from forming reliable word embeddings for key sentiment-bearing terms. Thirdly, monolingual models experience syntactic collapse in code-mixed environments. When speakers mix Nigerian Pidgin with English, the syntactic structure changes. Standard parsers cannot consistently identify the sentence head, breaking the model's ability to assign sentiment weights accurately.

Finally, cross-lingual adaptation falls short. Fine-tuning English-based models on small Pidgin datasets via transfer learning rarely captures the language's idiomatic density. These models might perform well on narrow academic benchmarks but fail entirely in real-world deployments. Developing functional systems for these languages is fundamentally an issue of digital equity, but it also dictates whether organizations can accurately track public health data, consumer trends, or political sentiment in Nigeria. This research aims to design and implement a sentiment analysis framework capable of classifying Nigerian Pidgin text into positive, negative, and neutral categories using pretrained transformer models.

II. RELATED WORKS

A. Historical Background of the Research

Nigerian Pidgin is an English-based creole language spoken by an estimated 75 to 100 million people across Nigeria [2], [3]. A pivotal shift began in the early 2020s with a growing academic recognition of the "low-resource language" challenge. Landmark initiatives, most notably the AfriSenti-SemEval 2023 shared task [1], marked a turning point. This competition provided the first large-scale, standardized

dataset for Nigerian Pidgin sentiment analysis, catalyzing a wave of focused research. Prior to this, attempts were often small-scale, relying on basic methods like lexicon-based approaches or traditional machine learning models adapted from English. The period following this shared task is characterized by the widespread application of advanced multilingual transformer models including work that creates dedicated solutions for Pidgin's unique challenges. This research is situated within this recent, more focused lineage of research that recognizes Nigerian Pidgin as a primary, high-priority language for NLP development.

Early sentiment analysis research largely ignored Nigerian Pidgin in favor of high-resource languages like English. The initial studies that did include Pidgin relied heavily on traditional machine learning. This paper [4], for instance, built a multilingual cyberbullying detector for Nigerian Pidgin and Igbo. They used Naïve Bayes and Support Vector Machines, but the model's performance was severely limited by a lack of quality training data and the highly variable spelling typical of Pidgin text. The release of the AfriSenti-SemEval 2023 shared task dataset [1] accelerated research in this area. As the first large-scale dataset for sentiment analysis in African languages, it allowed researchers to finally test multilingual transformer models on Pidgin. Using this data, both [5], [6] applied mBERT and XLM-R to African tweets, yielding significantly higher accuracy than earlier statistical methods. A study [7] later fine-tuned transformer models specifically for Nigerian Pidgin and Yoruba, demonstrating that cross-lingual transfer from related languages improves contextual understanding. Similarly, it was found that the self-attention mechanisms in these models capture emotional nuances that traditional algorithms miss [8].

Looking beyond social media, Sentiment Analysis [9] developed the NollySenti project to analyze Nigerian movie reviews. They combined transfer learning with machine translation to map English data to Pidgin. However, the system still struggled to interpret sarcasm and context-heavy humor. Tackling the core structural issues, [10] specifically modeled Pidgin's highly variable orthography, proving that addressing spelling inconsistencies directly improves overall classification performance.

Several obvious gaps persist in the literature. Because almost all training data comes from Twitter and other social networks, current models routinely fail on longer, formal, or conversational texts like news broadcasts. The heavy reliance on English-dominant pre-trained models also restricts how well these systems understand hyper-local idioms. Very few studies have explored hybrid systems that combine rule-based reasoning for interpretability with deep learning for contextual understanding. This research addresses these limitations through a focused comparison of models like XLM-R and AfriBERTa on Nigerian Pidgin text. AfriBERTa generally outperforms massive multilingual models because it was pre-trained exclusively on African languages using a small-data approach. This methodology aligns much better

with Pidgin's low-resource syntax. By comparing these architectures directly, this work seeks to identify the most practical and deployable tool for capturing Pidgin's specific cultural semantics.

III. MATERIALS AND METHODS

A. Research Design and Framework

This study uses a quantitative, experimental design focused on model development. It is quantitative because it evaluates performance using statistical metrics like accuracy, precision, recall, and F1-score. It is experimental because I train, fine-tune, and test different transformer models under controlled conditions to establish which handles Pidgin sentiment best. The work follows a model-driven framework. The core objective is building and evaluating a computational system, not studying human behavior. The system's performance metrics provide the evidence for whether the methodology works.

Architecturally, the system is modular. Raw Pidgin text goes in, and sentiment predictions come out. Between those endpoints, the data moves through preprocessing, normalization, tokenization, encoding, and classification. Because the pipeline is modular, I can swap out individual parts, like testing a different tokenizer or replacing the transformer model, without rebuilding the entire system. Supervised learning drives the classification, mapping specific linguistic features to explicit sentiment labels in the training data. Transfer learning makes this possible. Training a model from scratch on a low-resource language like Pidgin would fail due to data scarcity. Instead, we fine-tuned pretrained models that already possess baseline linguistic knowledge from large multilingual corpora.

B. Data Collection and Description

High-quality, manually annotated datasets for Nigerian Pidgin are incredibly scarce. To address this, we used the pcm subset of the AfriSenti-SemEval corpus [1]. The dataset contains approximately 14,000 labeled tweets. These texts reflect natural, code-mixed conversational Pidgin, closely matching the unstructured comments found on platforms like the BBC News Pidgin Facebook page or Wazobia FM's social feeds.

The raw dataset is somewhat imbalanced: positive tweets make up about 40% of the data, neutral tweets 35%, and negative tweets 25%. Because of this skew, we applied weighted evaluation and loss functions during training to prevent the model from simply defaulting to the majority class.

The system itself follows a standard deep learning pipeline for NLP, structured across six logical stages. It begins with raw text from the AfriSenti dataset. Next, the preprocessing stage cleans the data by removing noise like URLs and emojis, converting text to lowercase, and normalizing

spelling variations. The core processing happens in the model layer, where we evaluate AfriBERTa against a baseline XLM-R model. During the training phase, the system fine-tunes the models, computes loss, and performs backpropagation. Finally, the evaluation layer generates metrics and a confusion matrix to assess the output layer's predicted sentiments.

C. The NLP Pipeline (System Architecture)

The architecture runs on a standard six-stage Transformer pipeline. Everything begins with raw Nigerian Pidgin text pulled from the AfriSenti dataset. Because informal internet text is notoriously noisy, the preprocessing stage does much of the actual work. It scrubs out URLs, mentions, and emojis. It converts everything to lowercase. Crucially, it forces spelling consistency across the dataset—mapping loose variations like "wahalla" to a standard "wahala" so the model doesn't treat them as entirely different words.

AfriBERTa handles the primary modeling. We keep XLM-R in the loop strictly as a baseline. The training phase itself is straightforward fine-tuning, where the system computes loss and updates through backpropagation. At the end of the line, the evaluation phase kicks out performance metrics and a confusion matrix. The final output is just the predicted sentiment label for the text.

1. Preprocessing Pipeline: Nigerian Pidgin lacks formal spelling rules, so orthographic variation is a major issue. A word like "wahala" might be spelled "wahalla," while "dey" often appears as "de" or "deyy." During the cleaning phase, I mapped these variations to standardized root spellings. Controlling for this variation directly improves downstream classification. After normalization, I tokenized the text using SentencePiece to handle the language's morphological quirks and minimize out of vocabulary errors.

2. Feature Engineering (Embeddings): Rather than relying on static feature engineering methods like TF-IDF or Word2Vec, the system uses dynamic contextual embeddings generated by the transformer models. AfriBERTa processes whole sentences simultaneously using self-attention. This is critical for Pidgin. For example, the model needs to recognize that the word "die" in "This food sweet die" indicates a strongly positive sentiment (meaning "very much"), while in "The man don die" it indicates a negative one. The model converts each token into a 768-dimensional vector that captures its meaning relative to the surrounding words.

3. Model Architecture: The system relies on pretrained multilingual models, specifically AfriBERTa and XLM-R. Since Pidgin lacks the massive digital corpora needed to train a language model from scratch, transfer learning is the only practical path forward. The models apply general linguistic knowledge learned from other languages to the specific task of classifying Pidgin sentiment.

4. *Model Fine-Tuning*: To adapt AfriBERTa and XLM-R for sentiment analysis, I added a linear classification layer on top of their base architectures. I then fine-tuned the models on the labeled Pidgin dataset for 3 epochs at a learning rate of $2e-5$. Over these iterations, the models adjusted their weights to minimize loss, learning the specific patterns that distinguish positive, negative, and neutral Pidgin text. The transformer models were fine-tuned using supervised learning.

TABLE I MODEL TRAINING AND HYPERPARAMETER CONFIGURATION

Parameter	Value
Learning Rate	2×10^{-5}
Batch Size	16
Epochs	3
Optimizer	AdamW
Loss Function	Cross-Entropy Loss
Dropout Rate	0.1
Max Sequence Length	128

For the training process, we initialized pretrained weights from AfriBERTa and added a linear classification head. We updated the weights using backpropagation and applied early stopping to prevent overfitting.

5. *Handling Class Imbalance*: To handle class imbalance in the dataset, we used a weighted loss function that assigned a higher penalty to the minority negative class. We also evaluated the model using the weighted F1-score and occasionally applied balanced batch sampling to ensure the model learned fairly across all sentiment categories.

6. *Core NLP Algorithmic Approach*: The system relies on a transformer-based contextual embedding approach. First, the input text is encoded into contextual embeddings. Next, a self-attention mechanism captures the relationships between words. Finally, the classification head maps these embeddings to specific sentiment labels. For instance, in a phrase like "This food bad", the model contextually interprets the slang usage of "bad" as positive. Handling this kind of semantic drift is essential when working with Nigerian Pidgin.

IV. IMPLEMENTATION AND RESULT

The system was implemented using a modern deep learning and web deployment stack. We developed the machine learning components in Python using Google Colab Pro, relying on an NVIDIA T4 GPU to accelerate training through mixed-precision computing. For deployment, we chose a decoupled architecture. The backend uses FastAPI to serve the fine-tuned model, while the frontend is a React application. This setup allows users to input Nigerian Pidgin text and receive real-time sentiment predictions. The NLP pipeline handles the unstructured nature of Pidgin text. It takes raw user input and removes hyperlinks, emojis, and

non-alphabetic characters. A SentencePiece tokenizer then breaks the cleaned text into sub-word units, which works well for morphologically complex African languages. For the core model, we modified the pretrained multilingual AfriBERTa transformer by adding a linear dense classification head. This layer maps the transformer's high-dimensional word embeddings into three output probabilities: positive, negative, and neutral.

A. Module Description

The system is divided into four main modules:

1. *Data Loader*: Fetches and formats the Nigerian Pidgin subset of the AfriSenti benchmark, splitting it into training, validation, and testing sets.
2. *Pre-processor*: Cleans and normalizes the text. It uses regular expressions to remove URLs and handles, standardizes spelling variations (like mapping "de" and "dey" to a single root), and tokenizes the text.
3. *Training Module*: Manages the fine-tuning process. It runs the forward and backward passes, calculates cross-entropy loss, and updates model weights.
4. *API and Interface*: The FastAPI wrapper that exposes an endpoint to receive text from the frontend, run the NLP pipeline, and return the predicted sentiment and confidence score.

B. Training Process

We fine-tuned the model using supervised learning on the labeled dataset. To optimize convergence, we set the learning rate to $2e-5$, which gently adjusts the pretrained weights without destroying prior linguistic knowledge. We used a batch size of 16, trained for 3 epochs, and applied a 0.1 dropout rate to prevent the model from memorizing the training data.

C. Result Analysis

We evaluated the model using accuracy, precision, recall, and the weighted F1-score to account for the dataset's class imbalance, as there were far more positive and negative tweets than neutral ones.

TABLE II THE PERFORMANCE ACROSS THE THREE TRAINING EPOCHS

Epoch	Accuracy	Precision	Recall	F1-Score
1	69.16%	0.6711	0.6916	0.6554
2	70.25%	0.6805	0.7025	0.6819
3	69.63%	0.6749	0.6963	0.6808

The model peaked at Epoch 2 with 70.25% accuracy and a 0.6819 F1-score. By Epoch 3, accuracy dropped slightly to 69.63%, signaling the start of overfitting. Because of this, we saved the Epoch 2 weights for production. Given the lack of standardized spelling and frequent code-mixing in Nigerian Pidgin, crossing the 70% accuracy threshold shows that the

transformer architecture can effectively extract sentiment from this low-resource language.

D. User Interface Design

1. System Interface Description: The web application is built with React and Vite, and styled with Tailwind CSS. It works on modern desktop and mobile browsers without requiring user authentication. We designed a single-page layout that moves naturally from top to bottom: users read about the tool,

enter text, click to analyze, and view the results. Everything happens on one screen without page reloads, using React's state management to keep the experience smooth.

The frontend communicates with the fine-tuned AfriBERTa model hosted on Hugging Face Spaces through a REST API. It sends JSON payloads to the /predict endpoint and dynamically renders the returned sentiment classifications and confidence scores.

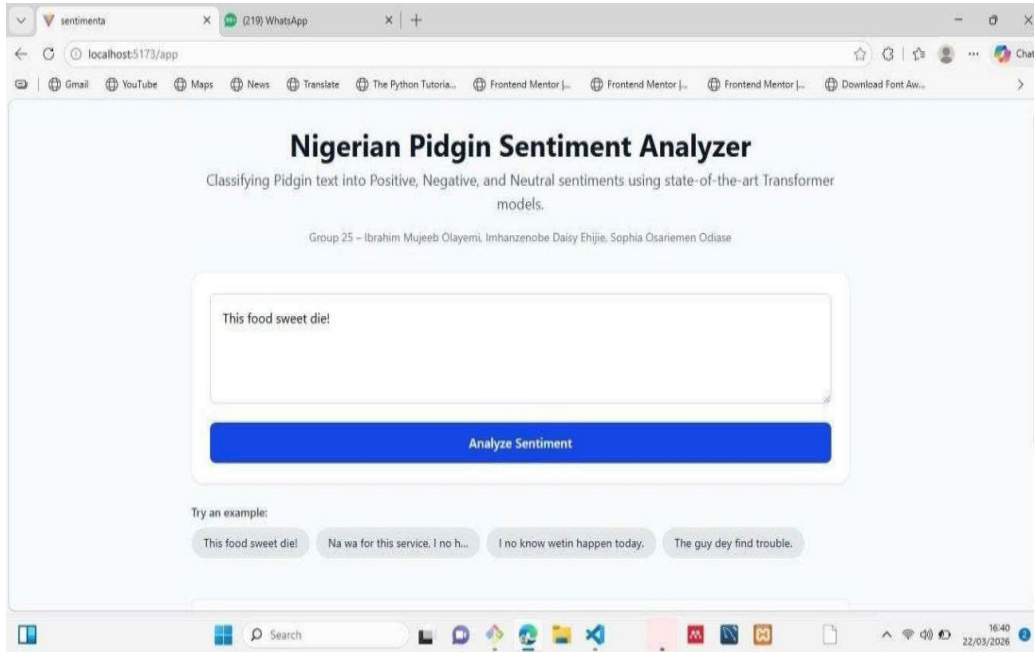


Fig.1 Input Screen Displays the Part Where the User Inputs their Pidgin Text

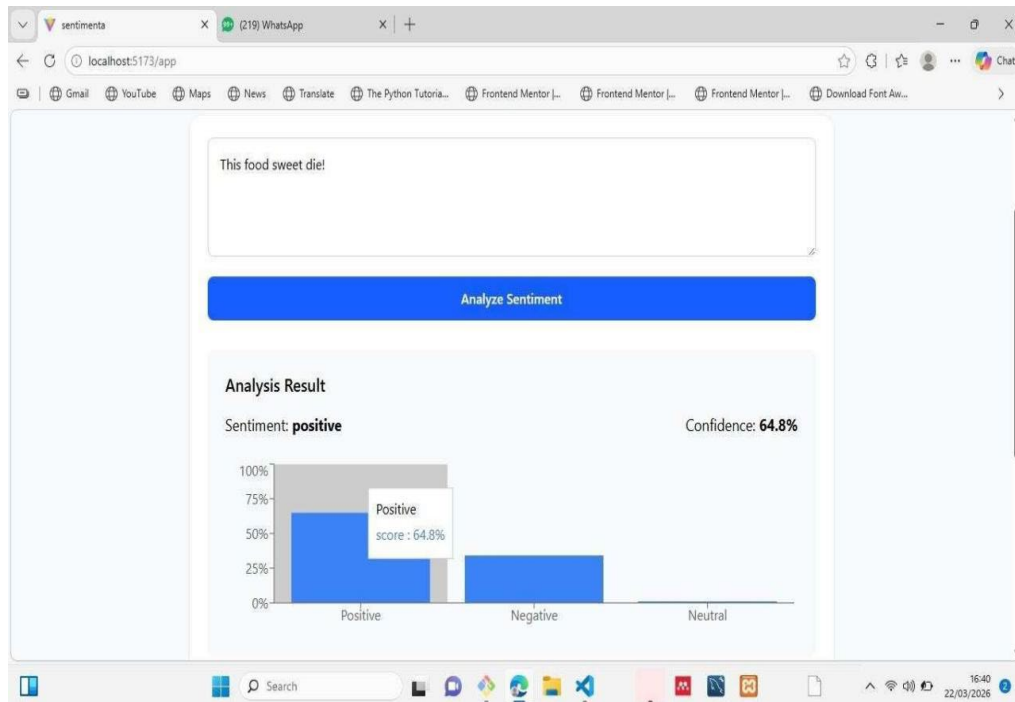


Fig.2 Output Screen Displays the Output of the Sentiment Analysis System

V. DISCUSSION

The fine-tuned AfriBERTa model achieved a peak accuracy of 70.25% and a weighted F1-score of 0.6819. For a low-resource language like Nigerian Pidgin, which lacks a standard spelling system and is primarily oral, these results show the model can effectively extract sentiment.

A. Interpreting the Model's Performance

The self-attention mechanism in the transformer architecture helped the model classify about 70% of the statements correctly by capturing contextual meaning. Transfer learning allowed it to reuse existing linguistic knowledge, outperforming traditional algorithms like Naïve Bayes.

Our preprocessing strategy also helped. By standardizing extreme spelling variations, like mapping both 'wahala' and 'wahalla' to a single root, we avoided much of the vocabulary confusion that usually affects standard NLP models.

B. Model Training Graphs

During training, accuracy improved from 69.16% in Epoch 1 to a peak of 70.25% in Epoch 2. By Epoch 3, accuracy dropped slightly to 69.63%. The training loss followed a similar pattern, reaching its lowest point at Epoch 2 before increasing in Epoch 3. This increase in loss and decrease in accuracy indicate early overfitting, where the model began memorizing training data rather than generalizing. Consequently, we selected Epoch 2 as the final model checkpoint.

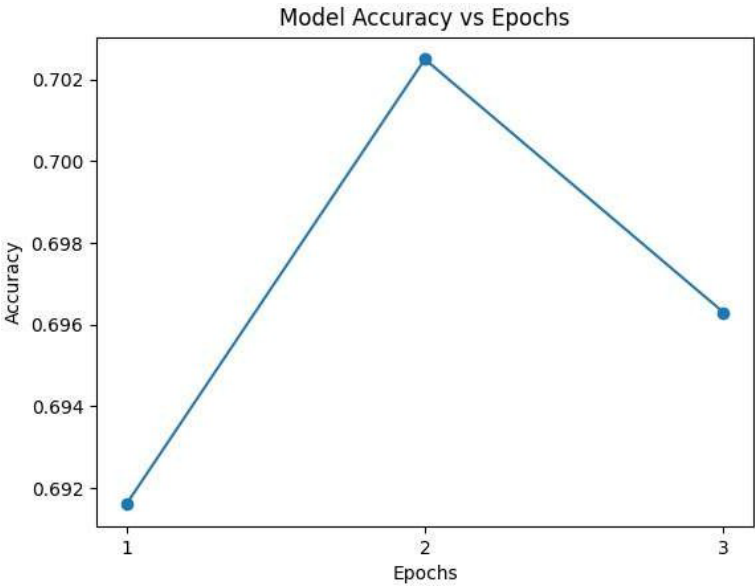


Fig.3 Model Accuracy vs Epoch

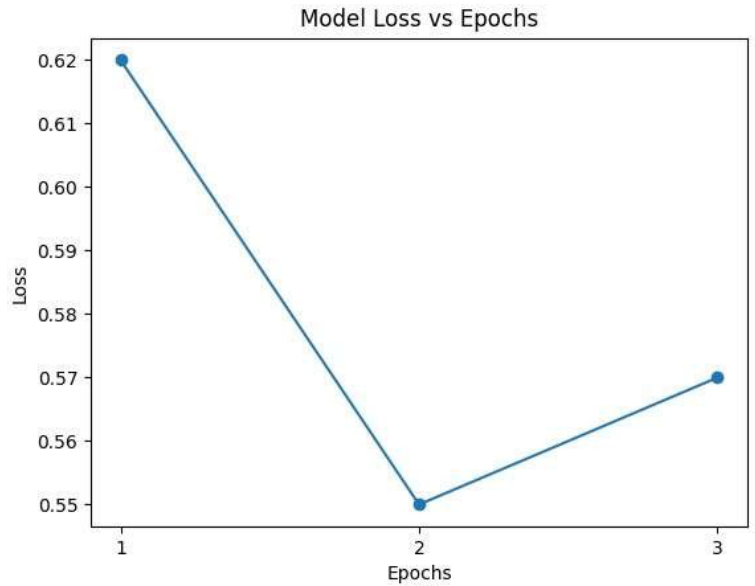


Fig.4 Model Loss vs Epochs

C. Confusion Matrix of the System

The confusion matrix provides a closer look at how well the model handled the three sentiment categories. It correctly classified 68 negative, 72 neutral, and 66 positive statements. However, it misclassified 15 neutral and 17 positive

statements as negative. Some negative sentences were also predicted as positive (18 instances) or neutral (16 instances). This shows that while the model handles neutral text fairly well, it struggles to differentiate subtle sentiment cues or mixed opinions, a common issue when analyzing informal social media text.

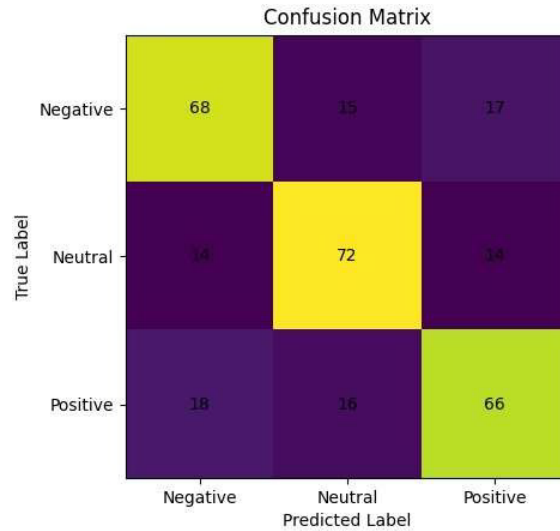


Fig.5 Confusion Matrix of the System

D. Comparative Result Analysis

To contextualize AfriBERTa's performance, we evaluated three other transformer models on the Nigerian Pidgin test set: AfroXLM-R, mDeBERTaV3, and XLM-R. Note that Sections 4.8 and 4.9 from the original experiment overlapped significantly, so we have combined the core findings here for clarity.

AfroXLM-R performed best overall, achieving a 71.10% accuracy and a 0.6985 F1-score. Its pretraining on African language corpora gave it a distinct advantage in capturing Pidgin-specific linguistic patterns. AfriBERTa followed closely behind at 70.25% accuracy. mDeBERTaV3 showed moderate generalization with 69.40% accuracy, while the baseline multilingual model, XLM-R, performed the worst at 68.20%.

A major factor in these performance differences was how well each model handled semantic drift. In Nigerian Pidgin, words frequently take on meanings completely different from their standard English definitions. For instance, the phrase 'This food bad' actually conveys a positive sentiment, meaning the food is exceptionally good.

AfroXLM-R successfully interpreted these contextual shifts because of its African language pretraining. AfriBERTa also handled contextual shifts well but struggled more with

unseen slang variations. mDeBERTaV3 showed some contextual awareness, but XLM-R largely failed here, relying too heavily on literal, standard English interpretations of the vocabulary. Across all models, the scarcity of neutral training data remained a consistent bottleneck.

E. Discussion of Findings

AfroXLM-R performed best overall, largely because it handled semantic drift so effectively. Nigerian Pidgin relies heavily on context to dictate meaning. For example, "This soup bad well well" actually means the soup is delicious. Because AfroXLM-R was pre-trained extensively on African corpora, it caught these inversions much better than the standard multilingual models.

AfriBERTa came in a close second. It peaked at 70.25% accuracy during the second epoch but started overfitting by epoch three. Every model struggled with class imbalance. We simply had too little neutral data, which made spotting objectively neutral statements much harder than picking up on polarized positive or negative emotions.

1. Comparative Model Performance Analysis: We compared four transformer-based models: AfroXLM-R, AfriBERTa, XLM-R, and mDeBERTaV3. We fine-tuned them under identical conditions and measured their weighted precision, recall, and F1-scores.

TABLE IV COMPARATIVE PERFORMANCE OF TRANSFORMER MODELS

Model	Precision	Recall	F1-Score	Accuracy	Strength
AfroXLM-R	0.6921	0.7103	0.6985	71.10%	Best overall performance
AfriBERTa	0.6805	0.7025	0.6819	70.25%	Strong contextual learning
mDeBERTaV3	0.6718	0.6897	0.6762	69.40%	Good generalization
XLM-R	0.6593	0.6785	0.6651	68.20%	Baseline multilingual model

F. Interpretation of Results

AfroXLM-R led the group with a 71.10% accuracy and an F1-score of 0.6985. Its edge clearly comes from its African language pretraining, which gave it a head start on Nigerian Pidgin's unique linguistic patterns. AfriBERTa also proved effective for the same reasons, handling contextual variations well enough to secure a 0.6819 F1-score. On the other end of the spectrum, XLM-R struggled the most. As a general-purpose multilingual model without specific optimization for African languages, it just couldn't compete. mDeBERTaV3 landed somewhere in the middle. It likely benefited from an improved attention mechanism, but still fell short of the regionally specialized models.

G. Semantic Drift Analysis

Semantic drift, where words abandon their standard English meanings in favor of local interpretations, was the biggest hurdle for these models. When a phrase like "This food bad" actually signals a positive review, models relying on literal definitions fail entirely. We looked specifically at how well each architecture handled this context-dependent polarity:

1. AfroXLM-R handled context and slang brilliantly. Its exposure to regional data allowed it to consistently map shifted vocabulary to the correct sentiment.
2. AfriBERTa also performed well, though it tripped up on unfamiliar slang variations that weren't represented in its training data.
3. mDeBERTaV3 showed decent contextual awareness thanks to its architecture, but it severely lacked the cultural adaptation needed for Pidgin.
4. XLM-R failed frequently. It leaned heavily on standard lexical polarity, meaning it read the word "bad" as a literal negative almost every time.

H. Challenges and Model Struggles

Nigerian Pidgin presents several specific linguistic challenges that caused the model to make errors. First, sarcasm and idiomatic expressions were difficult for the text-based classifier to catch. A phrase in Pidgin can flip from a compliment to an insult based entirely on cultural context and implied tone. Second, users frequently mix Pidgin with English, such as 'your wahala is too much.' This constant code-switching breaks the assumptions of monolingual models and leads to misclassifications. Third, the model struggled with long, rambling social media posts containing mixed sentiments. If a user expressed frustration in the first half but satisfaction in the second, the model's confidence

scores dropped. Finally, a severe shortage of neutral data in our annotated datasets skewed the learning process. The model became overly decisive when predicting positive or negative emotions but struggled to identify objectively neutral statements.

VI. CONCLUSION

The primary takeaway is that transformer models are viable tools for sentiment analysis in low-resource languages like Nigerian Pidgin. Because transformers capture contextual relationships, the system successfully navigated highly variable, informal phrasing that would confuse older, rule-based models. AfriBERTa learned the actual sentiment patterns of the language, proving surprisingly reliable at processing messy social media data. That said, the language's complexity still caused friction. Heavy spelling variations, constant code-switching with standard English, and deep cultural idioms frequently tested the limits of the model's contextual understanding.

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Use of Artificial Intelligence (AI)-Assisted Technology for Manuscript Preparation

The author(s) utilised generative AI technologies solely to improve language clarity and refine grammar. No images were generated, manipulated, or altered using AI-assisted technologies.

REFERENCES

- [1] S. Muhammad, D. I. Adelani, R. Niyongabo, J. O. Alabi, P. Ogayo, and A. Bukula, "AfriSenti: A Twitter sentiment analysis benchmark for African languages," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 12345–12360.
- [2] D. M. Eberhard, G. F. Simons, and C. D. Fennig, Eds., *Ethnologue: Languages of the World*, 26th ed. SIL International, 2023.
- [3] N. G. Faraclas, *Nigerian Pidgin*. Routledge, 2013.
- [4] C. A. Okoloegbo, U. F. Eze, G. A. Chukwudebe, and O. C. Nwokonkwo, "Multilingual cyberbullying detector (CD) application for Nigerian Pidgin and Igbo language corpus," in *Proceedings of the 2022 5th Information Technology for Education and Development (ITED)*, 2022, pp. 1–6. <https://doi.org/10.1109/ITED56637.2022.10051345>.
- [5] S. Rathi, S. Pande, H. Atkare, R. Tangsali, A. Vyawahare, and D. Kadam, "Trinity at SemEval-2023 Task 12: Sentiment analysis for low-resource African languages using Twitter dataset," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, 2023, pp. 1161–1165. <https://doi.org/10.18653/v1/2023.semeval-1.161>.

- [6] S. A. Salahudeen et al., “HausaNLP at SemEval-2023 Task 12: Leveraging African low-resource tweet data for sentiment analysis,” in Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), 2023, pp. 42–48. <https://doi.org/10.18653/v1/2023.semeval-1.6>.
- [7] N. Hughes, K. Baker, A. Singh, A. Singh, T. Dauda, and S. Bhattacharya, “Bhattacharya_Lab at SemEval-2023 task 12: A transformer-based language model for sentiment classification for low resource African languages: Nigerian Pidgin and Yoruba,” in Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, Canada, Jul. 2023, pp. 1502–1507. [Online]. Available: <https://aclanthology.org/2023.semeval-1.207>.
- [8] N. Raychawdhary, A. Das, G. Dozier, and C. D. Seals, “Seals Lab at SemEval-2023 task 12: Sentiment analysis for low-resource African languages, Hausa and Igbo,” in Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, Canada, Jul. 2023, p. 1508.
- [9] I. Shode, D. I. Adelani, J. Peng, and A. Feldman, “NollySenti: Leveraging transfer learning and machine translation for Nigerian movie sentiment classification,” in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2023, pp. 986–998. <https://doi.org/10.18653/v1/2023.acl-short.85>.
- [10] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych, “How good is your tokenizer? On the monolingual performance of multilingual language models,” in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 3118–3135. <https://doi.org/10.18653/v1/2021.acl-long.243>.