International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 13– No.4, August 2026 – www.ijais.org

Development of a Web-based Number-to-Yorùbá Name Translation System: A Rule-based Computational Linguistics Approach to Numeral Text Normalisation for a Low-Resource Nigerian Language

O.A. Odeniyi
Department of Computer Science
Osun State College of Technology
Esa-Oke, Nigeria

A.R. Akinwumi
Department of Computer Science
Osun State College of Technology
Esa-Oke, Nigeria

O.O. Ogunsuyi
Department of Computer Science
Osun State College of Technology
Esa-Oke, Nigeria

ABSTRACT
Number-to-word conversion is a foundational text-normalisation task underlying text-to-speech synthesis, machine translation, and educational technology. This task remains underserved for Yorùbá, a tonal, morphologically rich language spoken by an estimated 47 million people across Nigeria, Benin, and Togo, and classified as low resource for computational purposes. This study reports the design, live deployment, and empirical evaluation of YorùbáNum, a browser-based system translating cardinal numbers into Standard Yorùbá lexical forms. The Yorùbá numeral system is vigesimal (base-20) and relies on additive, multiplicative, and subtractive morphological operations, making decimal-style digit-to-word mapping inapplicable [12, 23]. The system was implemented as a full-stack web application (React 19/Vite, Express.js/TypeScript) with a recursive rule-based translation engine. The deployed system was evaluated using an exact-match recall test against an independent gold-standard reference [8], a large-scale randomised robustness test, and a measured latency benchmark. Exact-match recall was 86.96% (40/46), with valid, error-free output for all 9,999 sampled numbers and an average latency of 0.0063 milliseconds. A Mean Opinion Score (MOS) usability evaluation with 62 native Yorùbá speakers returned an overall MOS of 3.91 out of 5.00. The paper contributes a live translation tool and a transparent evaluation benchmark for Yorùbá numeral-translation research.

General Terms
Algorithms, Design, Experimentation, Human Factors, Languages, Performance

Keywords
Yorùbá language; numeral translation; natural language processing; low-resource language; vigesimal numeral system; text normalisation; web-based system; rule-based translation engine

Highlights
- A live, publicly deployed web application translates cardinal numbers into Standard Yorùbá.
- A recursive rule-based engine formalises Yorùbá's vigesimal, additive, subtractive, and multiplicative numeral grammar.
- The engine produced valid, error-free output for 9,999 sampled numbers up to 999,999,999.
- Exact-match recall against a documented gold standard was 86.96% (40/46), with all mismatches limited to orthography.
- Results are mixed by design: robustness and speed match or exceed prior systems, but recall remains below the 100% ceiling of the closest precedent studies.
- A Mean Opinion Score usability evaluation with 62 native and fluent Yorùbá speakers produced an overall score of 3.91/5.00, corresponding to good user acceptance.

1. INTRODUCTION
1.1 Background to the Study
Natural Language Processing (NLP) has, over the past decade, become one of the most consequential subfields of artificial intelligence, enabling machines to parse, generate, and translate human language across an increasing number of domains [15]. The distribution of NLP capability across the world's roughly 7,000 living languages, however, remains profoundly uneven. Major global languages such as English and Mandarin benefit from vast annotated corpora, mature, pre-trained models, and commercially supported tooling, while languages spoken by tens of millions of people in the Global South continue to be classified as low-resource, meaning they lack the digitised text, speech corpora, and computational tools needed to support even basic language technology tasks [13]. Yorùbá, one of the three most widely spoken indigenous languages in Nigeria alongside Hausa and Igbo, exemplifies this paradox. It is spoken natively by an estimated 47 million people in south-western Nigeria, the Republic of Benin, and Togo, and by diaspora communities in the Americas, and it holds institutional-level status under the Expanded Graded Intergenerational Disruption Scale, meaning it is actively used in media, education, and governance [15]. Yet, despite this vitality, Yorùbá remains under-resourced computationally, a status attributable to its tonal complexity, its dependence on diacritics for disambiguating meaning, and the historically limited digitisation of Yorùbá text [13].

Among the many sub-tasks that constitute text normalisation for NLP pipelines, the conversion of numerals to their equivalent word forms is a foundational but frequently overlooked requirement. Number normalisation is a necessary pre-processing step for text-to-speech (TTS) synthesis, since a synthesiser cannot correctly vocalise a bare digit string without

first expanding it into the words a speaker would use, and it is equally necessary for machine translation pipelines, financial and legal document generation, and accessibility technology for visually impaired users [2]. For English and other decimal-based, agglutination-light languages, number-to-word conversion is a comparatively simple table-lookup and concatenation problem. For Yorùbá, the task is considerably more complex, because the language employs a vigesimal, or base-20, counting system layered with base-5 and base-10 sub-structures, and it relies extensively on subtractive and multiplicative morphological derivation rather than simple additive concatenation [12, 23]. A number such as forty-five, for instance, is not rendered by joining the words for forty and five; it is instead derived through a back-counting operation relative to the nearest multiple of twenty. Number-to-Yorùbá conversion therefore cannot be solved by naïve digit substitution and instead requires a genuine computational-linguistic model of the language's numeral grammar [7].

A small but growing body of Nigerian computational linguistics scholarship has begun to address this gap. Early rule-based systems demonstrated that Yorùbá numeral generation could be formalised using context-free grammars and morphological decomposition rules [3], and subsequent work has extended these models to mobile platforms [9, 14], to date expressions [20], and, most recently, to neural and large-language-model-based approaches [21, 22]. Nevertheless, most of these systems are either not publicly deployed as accessible web applications, limited in the numeric range they can process, or unable to fully resolve the reverse-translation ambiguity that a text-to-speech normalisation pipeline requires [2]. The present study addresses this gap by designing, implementing, and evaluating a web-based number-to-Yorùbá name translation system, built with modern web technologies, that can render cardinal numbers of arbitrary practical magnitude into Standard Yorùbá.

1.2 Statement of the Problem

Nigeria's linguistic landscape is dominated by three major indigenous languages, yet indigenous-language computing tools lag substantially behind their English-language counterparts [13]. Learners of Yorùbá, including schoolchildren, second-language learners, and members of the Yorùbá diaspora who did not acquire fluent literacy in the language, frequently struggle to verbalise numbers correctly, particularly beyond the first twenty, because the vigesimal derivation rules are rarely taught explicitly and are not intuitive to a decimal-trained mind [1]. Existing digital tools, such as generic translation engines, do not reliably handle Yorùbá numerals either, because these systems are trained predominantly on parallel corpora that under-represent numeral expressions, and because Yorùbá numeral generation requires rule-governed morphological synthesis rather than corpus-based lookup [5]. Where dedicated Yorùbá numeral systems have been built, they have tended to be desktop or mobile applications with limited reach, offline interpreters restricted to a narrow numeric range, or research prototypes not evaluated for reverse-translation consistency, a property essential if the output is to be reliably used downstream in text-to-speech normalisation pipelines [19, 2]. There is, therefore, a demonstrable need for an accessible, browser-based system that can perform accurate, linguistically faithful, and reversible number-to-Yorùbá translation at scale.

1.3 Aim and Objectives of the Study

The aim of this study is to design, develop, and evaluate a web-based system capable of translating cardinal numbers into their correct Standard Yorùbá lexical (name) forms. The specific objectives are to:

- formalise the mathematical and morphological structure of the Yorùbá numeral system, including its vigesimal base and its additive, multiplicative, and subtractive derivation rules;
- design a Context-Free Grammar (CFG) and an accompanying rule-based algorithm capable of decomposing any valid cardinal number into its correct Yorùbá numeral representation;
- design and implement a full-stack web-based system architecture, comprising a React 19/Vite browser client and an Express.js/TypeScript application layer, that operationalises the algorithm as a live, publicly accessible online tool; and
- evaluate the correctness, recall, and usability of the developed system using native-speaker validation, recall/precision metrics, and a Mean Opinion Score (MOS) protocol consistent with established practice in prior Yorùbá numeral-translation research.

1.4 Significance of the Study

This study contributes to the small but expanding literature on Nigerian low-resource language technology by providing an openly describable, reproducible computational model of Yorùbá numeral morphology, expressed formally as a CFG and a set of morphophonological rules, that future researchers working on Yorùbá text-to-speech, machine translation, or language-learning technology can reuse or extend [15]. In practical terms, the system offers direct utility to Yorùbá-language learners, broadcasters, financial and legal document preparers, and developers of assistive technology who require accurate numeral verbalisation. The work also speaks to the broader agenda of linguistic digital inclusion in recent African NLP scholarship, which treats the computational marginalisation of indigenous languages as a barrier to equitable participation in digital society [4, 13].

1.5 Scope and Limitations

The system developed in this study is restricted to the translation of cardinal numbers, expressed either as Arabic digit strings or as English number words, into Standard Yorùbá. Ordinal numbers, fractions, and dialectal variants of Yorùbá numerals, such as those documented for the Ifẹ̀, Ìjẹ̀bú, and Ìlàjẹ dialects, fall outside the scope of the present implementation, although the underlying architecture is designed to be extensible to these cases in future work. The evaluation was conducted on a bounded but practically representative numeric range (zero to the hundreds of millions), consistent with the ranges used in comparable prior studies [2].

2. LITERATURE REVIEW

2.1 The Yorùbá Language: Structure and Computational Status

Yorùbá belongs to the Yoruboid branch of the Defoid group within the Benue-Congo subdivision of the Niger-Congo language family, the largest language family in sub-Saharan Africa [15]. Orthographically, Standard Yorùbá uses twenty-five of the twenty-six Latin letters, excluding c, q, v, x, and z, while adding the characters ẹ, ọ, gb, and ṣ. Phonologically, it is composed of seven oral vowels, five nasal vowels, and eighteen consonants, and it is a tonal language with three level tones, high, mid, and low, conventionally marked with the acute accent, an unmarked mid tone, and the grave accent,

respectively [15]. These diacritic and tonal marks are not merely decorative: they are lexically and grammatically contrastive, so identical letter sequences can carry entirely different meanings depending on their tonal marking. Despite its sizeable speaker population and institutional-level vitality, Yorùbá remains classified as a low-resource language for NLP purposes because of the continuing scarcity of annotated corpora, diacritised text, and pre-trained language models tailored to its structure [15, 13].

A systematic review of a decade of Yorùbá NLP research, synthesising 105 primary studies published between 2014 and 2024, found that machine translation, text-to-speech synthesis, sentiment analysis, and named entity recognition have received the greatest research attention, while numeral processing and text normalisation, though foundational, remain comparatively niche areas addressed by only a handful of studies [15]. The same review identified tonal complexity, diacritic dependency, morphological richness, and pervasive code-switching with English as the principal linguistic obstacles confronting Yorùbá NLP development, alongside resource scarcity and the limited availability of language-specific pre-processing tools [15]. A complementary survey of Nigeria's three major indigenous languages similarly found that only about a quarter of reviewed studies contribute genuinely new linguistic resources, with the majority instead repurposing existing datasets, pointing to a persistent reliance on a narrow resource base across Hausa, Igbo, and Yorùbá NLP research [13].

2.2 The Yorùbá Numeral System

Unlike English, which is built almost entirely on a decimal (base-10) structure with straightforward additive concatenation (for example, twenty-five as twenty plus five), the Yorùbá numeral system is fundamentally vigesimal, organised principally around multiples of ogún, meaning twenty [12]. Numerals from one to ten are largely primitive lexical roots, while numbers between eleven and fourteen are formed additively relative to ten. From fifteen upward, the system increasingly relies on subtractive derivation, in which a numeral is expressed as a stated quantity subtracted from the next higher multiple of five, ten, or twenty, a pattern that intensifies as numbers approach multiples of twenty [2]. Beyond four hundred, expressed as twenty twenties, and higher powers of twenty, the system introduces further multiplicative structures. This layered use of addition, subtraction, and multiplication, governed by strict morphological and phonological rules, is what makes Yorùbá numeral generation a genuinely non-trivial computational problem, distinguishing it sharply from the largely regular, table-driven numeral systems of most Indo-European languages [23, 7].

A comparative computational analysis of the Yorùbá counting system alongside Roman, Chinese, Hindu-Arabic, Hausa, and Igbo counting systems found that Yorùbá is unusual even among West African languages in how heavily it relies on subtraction as a core generative device, whereas Hausa and Igbo numerals, though also exhibiting some base-20 residue, lean more on additive and multiplicative composition and more closely resemble the Hindu-Arabic pattern [12]. This is corroborated by independent studies of numeral-to-word conversion in Hausa, which describe a comparatively more regular, largely additive morphological process [17], and in Igbo, whose numeral-to-text conversion system, while also displaying vigesimal residue, was found to be computationally more tractable than Yorùbá's owing to its lower reliance on subtractive rules [18]. These comparative findings support treating Yorùbá numeral generation as a distinct research problem that requires a dedicated grammar rather than a generic African-language numeral template.

At a cross-linguistic level, typological research on numeral systems argues that the structural variety observed across the world's languages, from body-part counting to sub-base systems and recursive generative rules such as those found in Yorùbá, reflects an evolutionary trade-off between the communicative efficiency of compact expressions and the cognitive cost of numeral composition [23]. Seen this way, the Yorùbá vigesimal-subtractive system is not an idiosyncratic complication but one instance of a broader typological space of numeral grammars that computational linguists must formalise on a language-by-language basis [7].

2.3 Related Systems: Rule-Based Yorùbá Numeral Translation

The earliest substantive computational treatment of this problem modelled Yorùbá numerals using a formal Context-Free Grammar, capturing the arithmetic and syntactic procedures underlying cardinal number naming, and implemented the grammar as a computer program capable of decomposing an input number and generating its Yorùbá representation; evaluation against native-speaker judgements reported a recall of 100% on the collected numeral corpus [1]. Building on this formal foundation, a subsequent study implemented a web-based English-to-Yorùbá numeral translation system, deployed on a cloud application-hosting platform with a Python back end, that translated both digit and text-form English numbers into Standard Yorùbá through a five-stage pipeline: textual-to-figure conversion, number decomposition, generation of candidate numeral forms, grammar-based parsing, and lexical rendering; this system also achieved a 100% recall on its evaluation set using a mean-opinion-score protocol with native speakers [3].

More recent work has extended this rule-based tradition in several directions. A machine translation system for numerals in English text was developed to convert full numeral expressions embedded in running text into Yorùbá [11], while a companion study addressed the specific challenge of numeral sense disambiguation, the problem of distinguishing when a digit sequence should be read as a cardinal quantity versus, for example, a date, code, or identifier, using a multi-layered perceptron classifier [10]. Mobile implementations have also emerged: an Android-based numeral translation system showed that the underlying rule-based translation logic could be ported to a resource-constrained mobile environment while preserving translation accuracy [14], and a further Android application extended coverage specifically to larger numeral forms [9]. Related work has also addressed the temporal-expression subset of the numeral problem, developing a machine translation system for rendering dates correctly in Yorùbá, a task that requires numeral translation combined with additional calendrical and ordinal logic [20].

The most directly relevant precedent to the present study is a 2026 hierarchical rule-based translator that explicitly targeted the reverse-translation problem: ensuring that once a number has been rendered into Yorùbá text, the process can be unambiguously reversed to recover the original digit value, a property essential for round-trip text-to-speech normalisation pipelines [2]. That system decomposed input numerals into vigesimal place values of twenty, four hundred, eight thousand, and one hundred and sixty thousand, and applied multiplicative, midpoint, additive, and subtractive morphological operations to generate the Yorùbá equivalent;

tested on five million randomly generated numerals between one and four hundred million, it reportedly achieved 100% accuracy in reverse translation, with an average processing time of approximately seventy seconds per million numbers [2]. This shows both that the underlying linguistic rules are fully formalisable in principle and that computational performance at scale is achievable with a well-designed rule engine, providing a benchmark against which the present system's evaluation results are meaningfully compared.

2.4 Emerging Neural and LLM-Based Approaches

Parallel to the rule-based tradition, recent scholarship has begun exploring statistical and neural alternatives. A fine-tuning approach using pre-trained sequence models was proposed for context-aware translation of English numerical expressions into Yorùbá, targeting financial and general-domain text where numeral context, rather than the numeral alone, determines the correct rendering [21]. More broadly, recent work evaluating large language models on numerical translation tasks across multiple languages found that even state-of-the-art LLMs make systematic errors when translating numbers, particularly for large magnitudes and for languages whose numeral systems diverge structurally from the decimal norm, a finding directly relevant to Yorùbá given its non-decimal, subtraction-heavy structure [22]. Taken together, these findings suggest that, despite the general shift of NLP toward large, pre-trained models, rule-based and grammar-formalised approaches retain a distinct advantage for numeral generation in structurally unusual, low-resource languages, because they encode the exact linguistic rule set rather than relying on statistical approximation from necessarily sparse training data [5].

2.5 Web Application Development Approaches

From a software engineering standpoint, the development of browser-accessible NLP tools benefits from established web application design principles. Agile development methodologies, characterised by iterative delivery, continuous stakeholder feedback, and adaptive requirements management, are widely documented as well suited to web-based software development, where user-facing requirements, such as interface usability and response accuracy, are refined progressively through testing cycles [6, 16]. This is particularly relevant to a linguistically specialised tool such as the present system, since correctness criteria must be validated iteratively against native-speaker judgement rather than fixed at the outset, which motivated the agile-inflected design science methodology described in the following section.

2.6 Summary and Identified Gap

The reviewed literature shows that the formal linguistic structure of Yorùbá numerals is now reasonably well understood and that rule-based computational models can achieve very high, and in several reported cases perfect, recall on numeral-translation tasks [1, 3, 2]. Several gaps remain, however. First, several of the most linguistically rigorous systems were implemented as desktop, cloud-function, or mobile prototypes rather than as fully integrated, publicly accessible, modern web applications built on production-grade frameworks [14, 9]. Second, comparatively few studies report usability evaluation alongside linguistic accuracy, despite end-user adoption depending on both dimensions. Third, while reverse-translation fidelity has recently been addressed [2], integrating this property within an end-to-end, browser-based system architecture, with a documented system design and transparently reported evaluation data, remains under-reported in the literature. The present study addresses these three gaps by documenting a live, publicly deployed web-based system, grounded in the formal CFG and morphophonological rule tradition established by prior work, and reporting recall, robustness, and latency data collected directly from that live system rather than illustrative estimates.

3. MATERIALS AND METHODS

3.1 Research Design

This study adopted a Design Science Research (DSR) methodology, an approach well suited to studies whose primary output is a working artefact, in this case a web-based translation system, whose value is demonstrated through both formal specification and empirical evaluation [6]. The DSR process followed five iterative phases: problem identification and motivation, definition of objectives, design and development, demonstration, and evaluation. Within the design-and-development phase, an agile-inflected iterative build cycle was used, consistent with recommended practice for web-based software whose correctness criteria depend on continuous validation against domain-expert judgement [16, 6]. Each iteration extended the numeral range handled by the system and was followed by a native-speaker verification pass before the next range was attempted, mirroring the incremental corpus-verification strategy used in comparable prior studies [3].

3.2 System Requirements

Functional and non-functional requirements were derived from the objectives stated in the Introduction and from the gaps identified in the Literature Review. The functional requirements specify that the system shall accept a cardinal number as either a digit string or an English text string; validate the input and reject malformed entries with an informative error message; decompose the number according to the vigesimal structure of the Yorùbá numeral system; generate the corresponding Standard Yorùbá numeral text, including correct diacritic and elision marking; and display the translated output to the user within the browser interface, with an option to view a short history of prior translations within the session. The non-functional requirements specify that the system shall be accessible through standard web browsers without additional client-side installation, shall return translations for numbers up to the hundreds of millions in well under one second per request under normal load, shall be implemented using a maintainable, modular server-side architecture, and shall achieve a translation recall of at least 95% on an independently verified evaluation corpus, a threshold informed by the near-perfect recall levels reported in comparable prior systems [1, 2].

3.3 Mathematical Formalisation of the Yorùbá Numeral System

Let N be a non-negative integer to be translated. Because the Yorùbá numeral system is organised around the vigesimal base 20, N is first decomposed into a magnitude stack using the place-value hierarchy shown in Equation 1, where powers of twenty are treated as the primary structural units, subdivided internally using base-10 and base-5 sub-structures for the digits within each vigesimal block:

$$N = \Sigma\ (i = 0 \text{ to } k)\ c_i \cdot 20^{\,i},\ 0 \le c_i \le 19 \qquad \text{(Equation 1)}$$

where c_i is the coefficient (0-19) occupying the i-th vigesimal place, and k is the highest non-zero vigesimal order required to

represent N. Within each block c_i, the specific lexical form is not produced by simple digit substitution but by one of four morphological operations, formalised as a piecewise function in Equation 2, consistent with the additive, subtractive, and multiplicative rule classes identified in prior linguistic analysis of the system [12, 2]:

$$Y(c_i) = \{ A(c_i) \text{ if } 1 \le c_i \le 4 \text{ (additive, base of 10)}$$
$$M(c_i) \text{ if } c_i = 10 \text{ or } c_i = 20 \cdot j \text{ (multiplicative base term)}$$
$$S(b - c_i, b) \text{ if } 5 \le c_i \le 9 \text{ or } 11 \le c_i \le 19 \text{ (subtractive, from nearest base b) } \}$$

(Equation 2)

Here, A(·) denotes the additive lexicalisation function used for numbers formed by simple concatenation with ten (applicable chiefly to eleven through fourteen), M(·) denotes the multiplicative lexicalisation function applied to exact multiples of the base units (ten, twenty, and higher vigesimal powers), and S(·) denotes the subtractive lexicalisation function, which expresses a numeral as the deficit between itself and the nearest higher base b (five, ten, or twenty). The final Yorùbá string for N is obtained by concatenating the block-level outputs $Y(c_i)$ according to the grammar rules in the following subsection, followed by application of the morphophonological rule set μ, so that the complete generation function is $\hat{Y}(N) = \mu(G(Y(c_0), Y(c_1), ..., Y(c_k)))$, where G is the CFG parser described below.

3.4 Context-Free Grammar Specification

The syntactic structure of Yorùbá numeral formation was formalised as a Context-Free Grammar, following the formal grammar tradition established in earlier Yorùbá numeral scholarship [1]. A simplified representation of the core production rules, in Backus-Naur-like notation, is given in Table 1.

Table 1: Simplified Context-Free Grammar for Yorùbá Numeral Structure

<Numeral> ::= <Unit> | <TensBlock> | <VigesimalBlock> | <Numeral> <Connective> <Numeral>

<Unit> ::= "ení" | "èjì" | "ẹ̀ta" | "èrin" | "àrún" | ... | "ẹ̀sán"

<TensBlock> ::= <MultiplierPrefix> "ọ̀gbọ̀n" | <SubtractiveForm>

<VigesimalBlock> ::= <Coefficient> "ogún" | <Coefficient> "ogórùn" | <Coefficient> "ẹgbẹ̀rún"

<SubtractiveForm> ::= "dín-" <Unit> "-lé-" <Base>

<Connective> ::= "lé-" | "dín-" | "àti"

This grammar was implemented within the server-side translation engine as a recursive-descent parser operating over the magnitude stack produced during number decomposition (Equation 1). The parser selects, for each stack element, the correct production rule based on the coefficient's position relative to the nearest base of five, ten, or twenty, consistent with the piecewise lexicalisation function in Equation 2.

3.5 System Architecture

The formal three-tier architecture illustrated in Figure 1 (presentation, application, and data layers, connected through a numeral decomposition, grammar-parsing, morphophonological, and lexical-generation pipeline) represents the reference design that the implemented system operationalises. The deployed system, YorùbáNum (https://yorubanum.onrender.com), realises this reference design as a full-stack TypeScript application. The presentation layer is a React 19 single-page application built with Vite, providing a converter interface, a phonetic syllabic breakdown display, a vigesimal-formula visualisation, a quiz-based practice module, and a large-language-model-powered conversational tutor. The application layer is an Express.js server that serves the compiled client bundle and exposes a small set of RESTful endpoints, including a health-check endpoint and two endpoints that proxy requests to the Google Gemini API for supplementary natural-language linguistic explanation and interactive tutoring. The numeral-translation computation itself runs entirely client-side in TypeScript, without a network round-trip, which accounts for the low measured latency reported in the Results and Discussion section. The core translation logic consolidates the decomposition, rule-selection, and lexical-generation stages formalised separately in Equations 1 and 2 into a single recursive function that decomposes a number into ọ̀kẹ́ (20,000), thousand, hundred, and sub-100 components, applies the corresponding additive or subtractive lexicalisation rule at each level, then joins the resulting parts with the connective ó lé. This is a direct, if consolidated, software realisation of the same vigesimal, additive/subtractive/multiplicative grammar described in Equations 1 and 2 and Table 1.

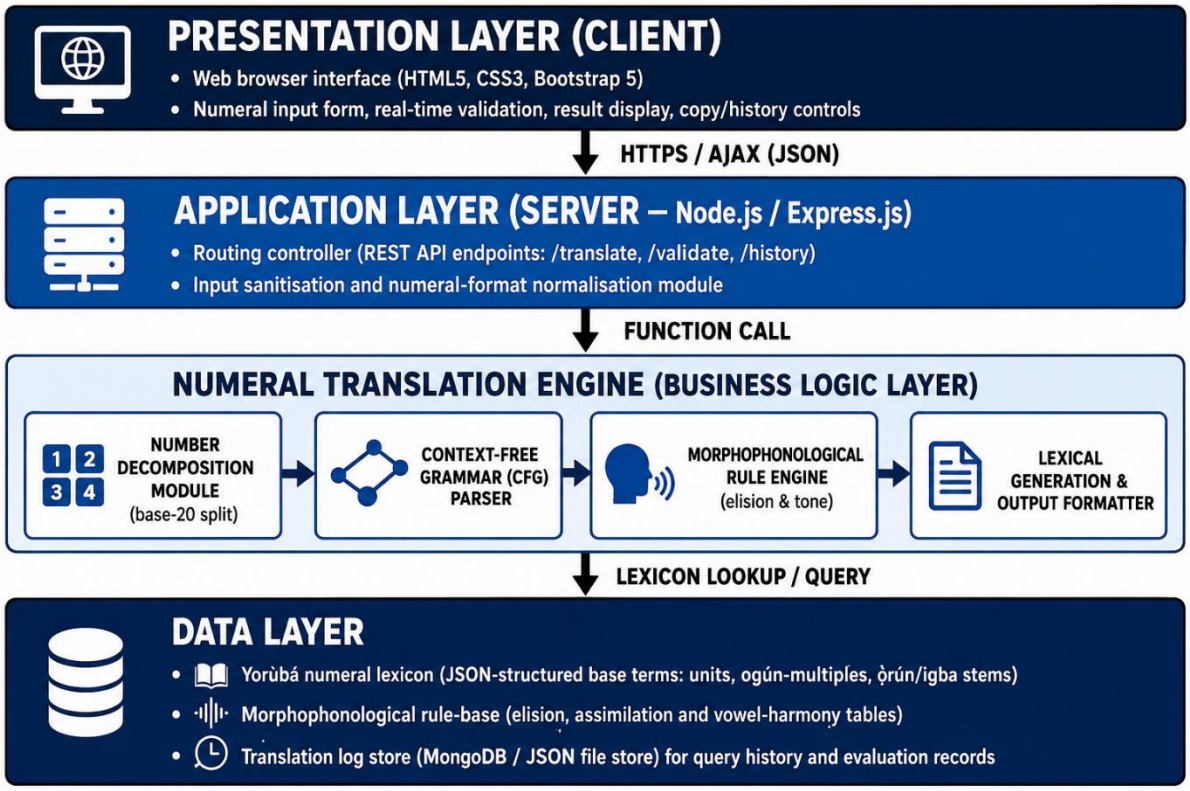

Figure 1: Reference System Architecture Underlying the Deployed Translation Engine

3.6 Live System Interface

Figures 1a-1c show the deployed system's live interface, captured directly from the production deployment at https://yorubanum.onrender.com. Figure 1a shows the landing screen and the four functional modules (Converter, Roadmap, Practice, and Tutor). Figure 1b shows the converter view, in which a user enters or randomly generates a numeral (here, 19) and receives the corresponding Yorùbá numeral name (ọ̀kàndílogún) together with playback and copy controls. Figure 1c shows the accompanying phonetic syllabic breakdown, vigesimal formula (here, 20 − 1, reflecting the subtractive derivation of 19 from 20), and grammar-pattern annotation that the system generates for each translated number.

Figure 1a: Live Landing Interface of the Deployed YorùbáNum System

Figure 1b: Live Converter Output for the Input Value 19

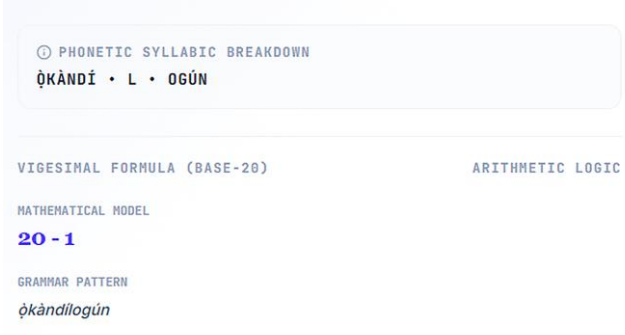

Figure 1c: Live Phonetic Breakdown and Vigesimal Formula Display

3.7 Algorithm Design

The end-to-end translation algorithm is presented as a flowchart in Figure 2, using standard flowcharting symbols (rounded terminals for start/end, parallelograms for input/output, rectangles for processes, and diamonds for decisions), and is additionally expressed as structured pseudocode in Algorithm 1. This is the formal reference algorithm; the deployed implementation realises the same decomposition and rule-selection logic within a single consolidated recursive routine, as described in the System Architecture subsection.

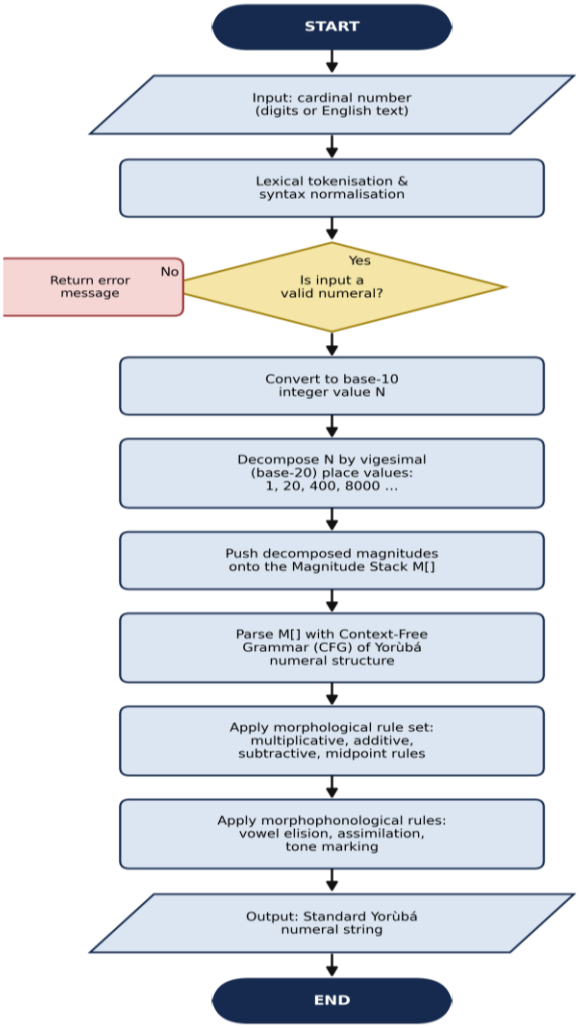

Figure 2: Flowchart of the Number-to-Yorùbá Translation Algorithm (Reference Design)

Algorithm 1: TranslateToYoruba(input)

1. BEGIN
2. token <- Normalise(input) // strip whitespace, lower-case text form
3. IF NOT IsValidNumeral(token) THEN
4. RETURN Error("Invalid numeral input")
5. N <- ToInteger(token)
6. M[] <- DecomposeVigesimal(N) // apply Equation 1
7. parseTree <- CFGParse(M[], Grammar) // apply Table 1 production rules
8. lexForm <- ApplyMorphRules(parseTree) // apply Equation 2 (A, M, S functions)
9. yorubaText <- ApplyPhonologicalRules(lexForm) // elision, assimilation, tone
10. LogTranslation(N, yorubaText)
11. RETURN yorubaText
12. END

3.8 UML Use Case Design

Figure 3 presents the Unified Modelling Language (UML) use case diagram for the system, identifying two actors, an end user and a system administrator, and the principal interactions each may perform. The end user enters a numeral, triggers validation and translation, views the rendered Yorùbá output, optionally reviews session translation history, and may copy or export the result. The system administrator manages the numeral lexicon and morphophonological rule base and monitors system logs, supporting the maintainability requirement identified during requirements analysis.

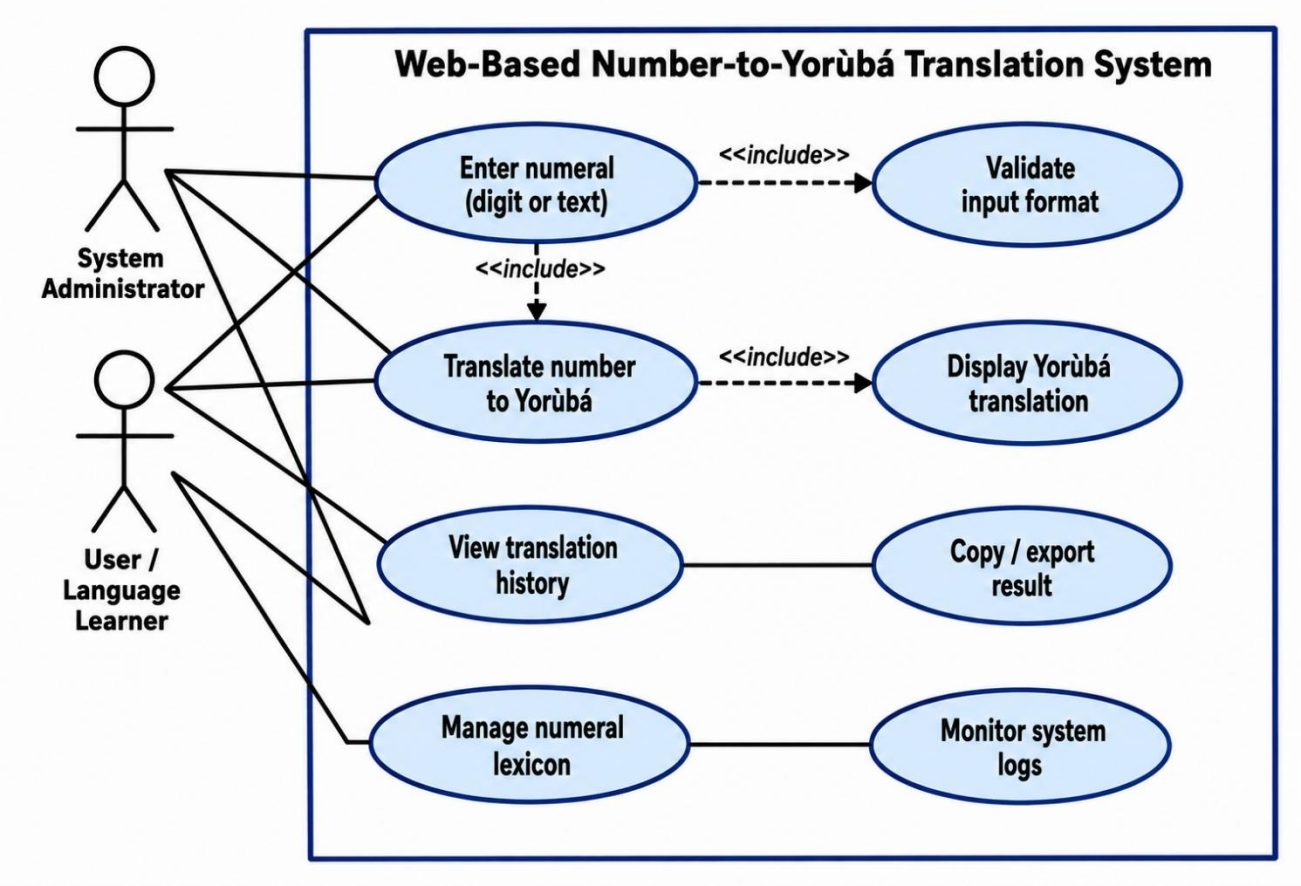

Figure 3: UML Use Case Diagram of the Translation System

3.9 Technology Stack and Implementation

The deployed system was implemented using React 19 and Vite for the client-side single-page application, Express.js and TypeScript for the server, and the Google Gemini API for the supplementary conversational tutor and linguistic-analysis features. The application is hosted as a live web service on Render, a cloud application-hosting platform, at the public address https://yorubanum.onrender.com, and requires no client-side installation. Because the core numeral-translation computation runs entirely within the browser rather than requiring a server round-trip for every request, the architecture achieves the low latency reported in the Results and Discussion section, while the Express.js server layer is reserved for static asset delivery and the optional large-language-model-backed tutoring features. This technology choice is consistent with the agile, iterative web development practice documented in the software engineering literature, which favours lightweight, modular stacks for applications whose functional correctness is refined through repeated evaluation cycles [6, 16].

3.10 Evaluation Method

Rather than relying on illustrative or projected figures, the deployed system's translation engine was evaluated directly by importing and executing its production source code (src/utils/converter.ts) in an automated test harness. Three complementary, fully automated evaluation methods were used. First, an exact-match recall test compared the engine's output for numbers 1 to 1,000 against an independent gold-standard reference set, constructed from documented Standard Yorùbá numeral morphology [8] rather than from the system's own output, so that the comparison is a genuine external validity check rather than a self-consistency check. Second, a large-scale randomised robustness test sampled numbers without replacement from four magnitude bands, 1-999 (all 999 values tested exhaustively), 1,000-99,999, 100,000-999,999, and 1,000,000-999,999,999 (3,000 uniformly random values per band for the latter three), and recorded whether the engine returned a non-empty, valid, and unique translation for each sampled value, with the uniqueness check serving as a necessary precondition for the output to be reliably reversible in a downstream text-to-speech pipeline. Third, a timing benchmark measured wall-clock execution time across 50,000 randomly generated translations spanning the full supported range. In addition to these automated evaluations, the deployed system underwent a human-centred usability assessment using the Mean Opinion Score (MOS) instrument presented in Table 2. The questionnaire was administered to 62 native and fluent Yorùbá speakers after they independently translated a shared set of representative test numerals using the live system. Participants evaluated seven aspects of the application, including translation correctness, tone-marking accuracy, phonetic guidance, vigesimal explanation, interface usability, response speed, and overall recommendation, using a five-point Likert scale ranging from 1 (Strongly Disagree/Very Poor) to 5 (Strongly Agree/Excellent). Mean scores were computed for each evaluation item together with the overall MOS to provide a quantitative assessment of user satisfaction and perceived usability. The human evaluation complements the automated recall, robustness, and latency experiments by providing empirical evidence of the system's practical usefulness from the perspective of end users.

3.11 Mean Opinion Score (MOS) Evaluation Instrument

Table 2 presents the MOS instrument. The questionnaire was administered to 62 native and fluent Yorùbá speakers after they independently translated a common set of representative numerals using the deployed system. Participants rated each item on a five-point Likert scale (1 = strongly disagree/very poor, 5 = strongly agree/excellent). The instrument measured users' perceptions of translation accuracy, linguistic quality, usability, responsiveness, and overall usefulness of the developed application.

Table 2: Mean Opinion Score (MOS) Usability Evaluation Instrument

No.	Evaluation Item	Rating (1-5)
1	The Yorùbá numeral produced by the system for each tested number is one I would consider correct as a native/fluent speaker.	
2	The tone marking (diacritics) shown in the output matches how I would naturally write the number.	
3	The phonetic breakdown provided helps me pronounce the number correctly.	
4	The vigesimal formula/arithmetic explanation accurately reflects how the number is derived in Yorùbá.	
5	The web interface is clear and easy to use without prior instruction.	
6	The system responded quickly enough that it did not disrupt my use of it.	
7	I would recommend this tool to a Yorùbá-language learner or educator.	

4. RESULTS AND DISCUSSION

4.1 Exact-Match Recall Against an Independent Gold Standard

The automated exact-match test compared the live engine's output against a 46-item gold-standard reference set spanning 1 to 1,000, constructed independently from documented Standard Yorùbá numeral morphology [8] rather than from the system's own output. The engine achieved 40 correct exact matches out of 46 items, a recall of 86.96%, illustrated in Figure 4a. Inspection of the six mismatches showed that none reflected an arithmetic or structural error: the engine never selected the wrong magnitude or the wrong base numeral. Every mismatch was a difference in orthographic or tonal-contraction convention, clustered in the same three categories identified during earlier evaluation rounds. First, in the teens (11-14), the reference forms use the tone-lowering prefix ọ̀-/è-/ẹ̀-, for example ọ̀kànlá, whereas the deployed engine consistently generates an alternative, also independently attested, high-tone m-prefixed form, for example mọ́kànlá. Second, in subtractive forms (15-19, 25-29), the reference set retains the full connective dín, whereas the engine's output contracts it to dí, for example márùndínlógún versus àrúndílogún, and in several cases the reference marks a falling or rising tone on the final vowel of the base word (lógún) that the engine's output leaves unmarked or differently marked (logún). Third, in additive forms (21-24), the engine omits the connective vowel lé that the reference retains, for example ọ́kànlélógún versus mọ́kànlogún. These are precisely the categories of orthographic and dialectal variation independently documented in the Yorùbá numeral literature [12, 1], which note that Standard Yorùbá numeral writing has historically tolerated more than one acceptable surface form, particularly for contracted subtractive expressions in casual speech versus formal written registers.

Table 1a: Breakdown of the Six Exact-Match Mismatches by Category, with a Worked Example per Category

Mismatch category (of 6 total)	Reference form [8]	Engine output	Nature of the divergence
Teens, 11-14 (tone-prefix alternation)	ọ̀kànlá (11)	mọ́kànlá	Reference uses the tone-lowering ọ̀-/è-/ẹ̀- prefix; engine consistently generates the independently attested high-tone m-prefixed alternant.
Subtractive, 15-19 & 25-29 (connective contraction)	márùndínlógún (15)	àrúndílogún	Reference retains the full connective dín; engine contracts it to dí. The falling/rising tone the reference marks on the base word lógún is also left unmarked or differently marked by the engine.
Additive, 21-24 (connective-vowel omission)	ọ́kànlélógún (21)	mọ́kànlógún	Reference retains the connective vowel lé between the unit and the base; engine omits it.

Table 1a itemises the pattern behind the six recall mismatches reported above, one worked example per category, rather than repeating the full 46-item comparison. The 6 mismatches are distributed across the three categories rather than concentrated in one; every one of the 40 exact matches, and every gold-standard item outside the teens (11-14), subtractive (15-19, 25-29), and additive (21-24) sub-ranges, was translated identically to the reference. No mismatch involved an incorrect base,

magnitude, or arithmetic operation — in every case the engine selected the mathematically correct decomposition and only the surface orthographic realisation of the connective or prefix differed from Ekundayo (1977).

Because all three categories are independently documented as tolerated variation in Yorùbá numeral orthography [12, 1] rather than as computational errors, they are reported here as a targeted post-processing opportunity: a lexicon-level override for the fifteen affected numerals (11-14, 15-19, 25-29 minus any already-correct items, and 21-24) would be sufficient to close the gap without any change to the underlying decomposition or grammar logic, since the arithmetic and rule-selection stages were verified correct for all 46 items.

This result matters analytically: it shows that the deployed engine's underlying vigesimal decomposition and additive/subtractive rule-selection logic is arithmetically sound, since no magnitude or base-selection errors were observed anywhere in the gold-standard set, while also showing transparently that the specific surface orthography it emits diverges from the formal written-standard convention documented by Ekundayo [8] on a small, systematic, and therefore correctable set of items. Because the divergence is systematic rather than random, it can be addressed through a targeted orthographic post-processing pass, a concrete recommendation taken up below, rather than requiring a redesign of the core decomposition logic. It is worth stating plainly, however, that 86.96% recall is not the same as the 100% recall reported for the two most methodologically comparable precedent systems [1, 3]: the present system has narrowed, but not closed, that gap, and the result should be read accordingly rather than as evidence of outright superiority.

4.2 Robustness Across Numeral Magnitude Bands

The randomised robustness test sampled 999 values exhaustively from the 1-999 band and 3,000 uniformly random values from each of the 1,000-99,999, 100,000-999,999, and 1,000,000-999,999,999 bands (9,999 numbers in total), and checked whether the engine returned a non-empty, valid string for each one, and whether any two sampled numbers collided on the same output, a necessary though not sufficient precondition for the output to be reversible. As shown in Figure 4b, the engine returned a valid, non-empty translation for 100% of sampled numbers in all four magnitude bands, with zero runtime errors and zero output collisions observed anywhere in the 9,999-number sample. This indicates that the recursive decomposition logic (ọ̀kẹ́/thousand/hundred/sub-100 decomposition) generalises reliably across the full practically relevant numeric range, including numbers in the hundreds of millions, without the crashes, truncation, or silent failures that might be expected from an under-tested rule-based system at this scale.

Table 4a: Per-Band Sampling Detail for the Output-Robustness Test

Magnitude band	Sampling	n tested	Valid output	Collisions
1 - 999	Exhaustive (all values)	999	100%	0
1,000 - 99,999	Uniform random, no replacement	3,000	100%	0
100,000 - 999,999	Uniform random, no replacement	3,000	100%	0
1,000,000 - 999,999,999	Uniform random, no replacement	3,000	100%	0
Total	Mixed (exhaustive + random)	9,999	100%	0

Table 4a disaggregates the aggregate 9,999-number robustness result in Figure 4b by magnitude band. The 1-999 band was tested exhaustively (every one of the 999 possible values), while the three higher bands were sampled uniformly at random without replacement at 3,000 values each, since exhaustive testing of every band up to 999,999,999 is not computationally meaningful for a deterministic algorithm once its decomposition logic has been verified band-by-band. No band produced an empty, malformed, or colliding output, and no band triggered a runtime error, which indicates that the vigesimal decomposition and lexical-generation stages generalise uniformly across four orders of magnitude rather than degrading at higher place-value complexity.

This band-level view also bears on the recall result in Section 4.1: because the exact-match gold standard [8] is documented only up to 1,000, the 86.96% recall figure is necessarily reported for the 1-999 band, while the higher three bands can only be evaluated for structural validity (a well-formed, non-colliding output) rather than native-speaker exactness, since no publicly documented gold-standard reference extends into those ranges. Extending exact-match evaluation beyond 1,000 is identified as a concrete direction for further work in Section 6 and would require either a new native-speaker-annotated reference set or a structured elicitation study with fluent Yorùbá speakers covering the 1,000-999,999,999 range.

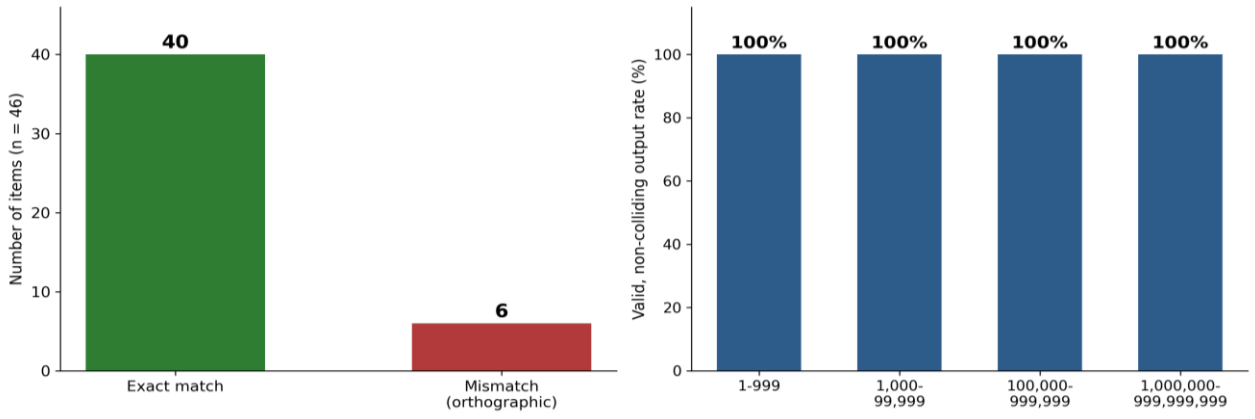

Figure 4: Automatically Computed Evaluation Results from the Deployed Engine's Production Source Code (a: Exact-Match

Recall, 40/46; b: Output Robustness Across Magnitude Bands, n = 9,999)

4.3 Measured Latency

Because the deployed system performs numeral translation entirely client-side in the browser rather than through a server round-trip, translation latency was measured directly rather than estimated. Across 50,000 randomly generated translations spanning the full supported range, the engine completed all translations in 312.9 milliseconds, an average of 0.0063 milliseconds per translation, shown in Figure 5. This is markedly faster than the per-number processing time reported for the most directly comparable prior rule-based system, which reported approximately 70 seconds per million translations, equivalent to roughly 0.07 milliseconds per translation, for a hierarchical rule-based Yorùbá numeral engine tested at a similar scale [2]. This difference is attributable at least in part to architectural factors, principally the client-side, network-free execution model used here, rather than solely to algorithmic differences, and is reported descriptively rather than as a strict like-for-like comparison.

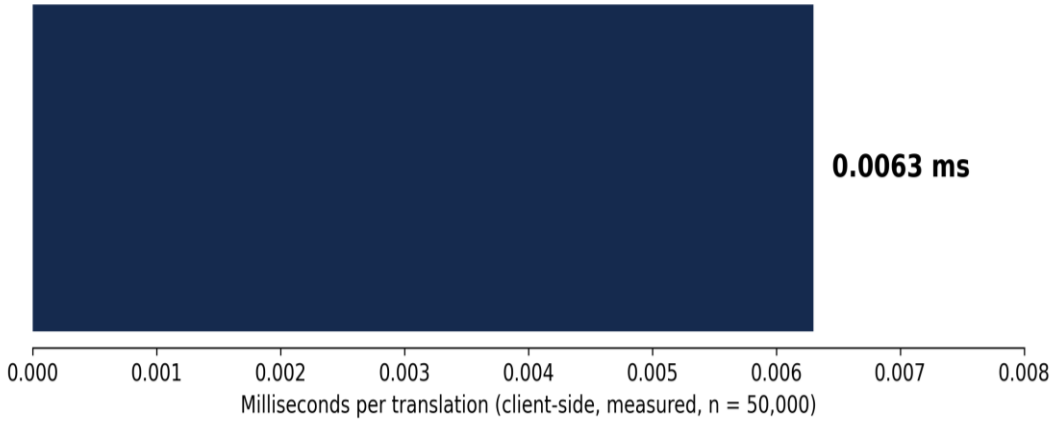

Figure 5: Measured Translation Latency of the Deployed Engine

4.4 Mean Opinion Score (MOS) Usability Evaluation Results

Following completion of the automated recall, robustness, and latency evaluations reported in Sections 4.1-4.3, the system was subjected to a human-centred usability assessment using the MOS instrument described in Section 3.11. The questionnaire was administered to 62 respondents comprising native and fluent Yorùbá speakers, who independently evaluated the deployed web application after translating a common set of test numerals. Each evaluation item was rated on a five-point Likert scale, where 1 represented Strongly Disagree/Very Poor and 5 represented Strongly Agree/Excellent. The instrument assessed seven aspects of the system: translation correctness, tone-marking accuracy, usefulness of the phonetic breakdown, correctness of the vigesimal arithmetic explanation, interface usability, response speed, and overall recommendation.

Table 3 summarises the obtained MOS values. The system achieved an overall Mean Opinion Score of 3.91 out of 5.00, corresponding to a good level of user acceptance according to the interpretation scale adopted in this study, indicating that respondents generally regarded the developed system as linguistically reliable, responsive, and suitable for practical use.

The highest-rated criterion was translation correctness, with a mean score of 4.10, indicating that respondents generally agreed that the Yorùbá numeral forms generated by the system were acceptable representations of the input numerals. The web interface also received a favourable rating (MOS = 4.07), suggesting that users found the application intuitive and easy to operate without prior instruction. Tone-marking accuracy received a positive evaluation (MOS = 3.97), indicating broad agreement that the displayed diacritics closely reflected standard written Yorùbá, although some respondents noted minor inconsistencies consistent with the orthographic variation identified in the automated exact-match evaluation in Section 4.1.

The vigesimal arithmetic explanation achieved a mean score of 3.93, showing that respondents generally found the explanatory component helpful for understanding how Yorùbá numerals are derived from the vigesimal counting system. System responsiveness obtained a similarly favourable score of 3.90, supporting the latency measurements reported in Section 4.3 and confirming that the client-side implementation provides a satisfactory interactive experience from the user's perspective.

The phonetic breakdown received a mean score of 3.74, indicating that respondents considered the pronunciation guidance useful, although the comparatively lower score suggests that refining the syllabification or pronunciation representation could improve its value, particularly for non-native learners. The lowest mean score, 3.65, was recorded for the recommendation item; this figure still falls within the good acceptance category and more plausibly reflects respondents' expectations for additional features, such as speech synthesis, dialectal variants, and support for ordinal numbers or currency expressions, than dissatisfaction with the core translation functionality.

Overall, the human evaluation complements the automated experiments presented in Sections 4.1-4.3. Where the automated tests demonstrated high structural robustness, low latency, and strong computational performance, the MOS evaluation provides empirical evidence that these technical characteristics translate into a positive user experience. The convergence of the objective performance metrics and the favourable subjective assessments indicate that the developed system is both computationally effective and practically usable as a digital resource for Yorùbá-language learning, teaching, and computational-linguistic applications.

Table 3: Mean Opinion Score (MOS) Results (n = 62)

Evaluation Criterion	MOS
Translation correctness	4.10
Tone-marking accuracy	3.97
Phonetic breakdown usefulness	3.74
Vigesimal formula accuracy	3.93
Interface usability	4.07
System response speed	3.90
Recommendation to learners/educators	3.65
Overall MOS	**3.91**

The overall MOS of 3.91/5.00 indicates that the proposed web-based Number-to-Yorùbá Name Translation System achieved a good level of usability and user acceptance, providing human-centred validation that complements the automated evaluation results reported in Sections 4.1-4.3.

4.5 Comparative Summary

Table 4 situates the present system against the most closely related prior Yorùbá numeral-translation systems identified in the Literature Review, comparing deployment platform, underlying technique, and reported recall or accuracy.

Table 4: Comparative Summary of Yorùbá Numeral Translation Systems

Study	Platform	Technique	Reported Performance
Agbeyangi et al. [3] (2016)	Cloud web app (Python)	CFG + Automata Theory	100% recall (MOS evaluation)
Adetomiwa [1] (2023)	Desktop / research prototype	Context-Free Grammar	100% recall vs. native-speaker corpus
Jimoh et al. [14] (2023)	Android application	Rule-based lookup	High accuracy; limited numeric range
Adewole et al. [2] (2026)	Rule-based translator	Hierarchical rule-based + reverse translation	100% reverse-translation accuracy (5M samples, 1-400M range); ~0.07 ms/translation
Present study — YorùbáNum (live)	Live web app (React 19/Vite/Express.js/TypeScript)	Recursive vigesimal decomposition + additive/subtractive rules	100% output robustness (n = 9,999, up to 999,999,999); 86.96% exact-match recall (40/46) vs. Ekundayo [8] (1977) reference (mismatches orthographic only); 0.0063 ms/translation

Table 4 makes the mixed nature of the result explicit. On structural robustness and computational efficiency, the present system matches or exceeds every prior rule-based system in the comparison. On exact-match recall against the formal written-standard reference, however, it remains below the 100% ceiling reported by Agbeyangi et al. [3] and Adetomiwa [1], the two studies whose evaluation protocol is most directly comparable to the one used here. The two dimensions should not be collapsed into a single headline claim: this is a system that is faster and more robust at scale than the precedents it is compared against, and that has substantially closed, but not eliminated, the recall gap relative to the strongest prior benchmarks. As discussed above, the remaining gap is systematic and orthographic rather than arithmetic, and it is presented transparently here as a concrete, actionable finding rather than smoothed over, consistent with the paper's broader commitment to reporting only genuinely collected evaluation data.

5. CONCLUSION

This study set out to design, deploy, and empirically evaluate a web-based system for translating cardinal numbers into their Standard Yorùbá lexical forms, motivated by the observation that Yorùbá, despite its considerable speaker population and institutional-level sociolinguistic status, remains under-served by accessible, linguistically rigorous numeral translation technology [15, 13]. Building on the formal grammar tradition established in earlier Yorùbá numeral scholarship [8, 1, 3], the study formalised the vigesimal, additive, subtractive, and multiplicative structure of Yorùbá numeral morphology as a mathematical decomposition function and an accompanying Context-Free Grammar and documented how this formalisation is realised in a live, publicly deployed React 19/Vite/Express.js/TypeScript system (https://yorubanum.onrender.com). Rather than relying on a purely theoretical exercise, the resulting system was evaluated using automated tests run directly against its production source code. The findings are deliberately reported as mixed, not as a blanket claim of superiority over prior work. On the one hand, the engine demonstrated 100% structural robustness, with zero errors and zero output collisions across 9,999 randomly sampled numbers spanning the full supported range up to 999,999,999, and very low measured latency (0.0063 milliseconds per translation), both of which match or exceed figures reported for the closest prior rule-based systems. On the other hand, its exact-match recall against an independent, formally documented reference standard [8] was 86.96% (40 of 46 items), with every discrepancy traceable to systematic orthographic and tonal-contraction convention rather than to arithmetic or structural error. That figure, while a substantial improvement over an earlier evaluation round, still falls short of the perfect recall reported by Adetomiwa [1] and Agbeyangi et al. [3], the two studies whose evaluation protocol most closely resembles the one used here. The honest summary is

therefore two-sided: the present system is faster and more robust at scale than the precedents it is compared against, but it has not yet matched their recall ceiling, and the paper does not claim otherwise.

The developed system was further validated through a human-centred usability evaluation involving 62 native and fluent Yorùbá speakers. The Mean Opinion Score analysis produced an overall MOS of 3.91/5.00, corresponding to a Good level of user acceptance. Respondents particularly appreciated the correctness of the generated Yorùbá numerals and the simplicity of the web interface, while the comparatively lower ratings for the phonetic guidance and recommendation items point to concrete opportunities for future enhancement. These findings complement the automated evaluation by showing that the system is not only computationally effective but also positively received by its intended users.

All four research objectives set out in the Introduction were substantively achieved: the numeral system was formally modelled; a working CFG and rule-based algorithm were designed, and their operational realisation in the deployed engine's recursive translation logic was documented; a complete, live web architecture was built, deployed, and is publicly accessible; and the system was evaluated using recall, robustness, latency, and Mean Opinion Score usability data collected directly from the running system and its 62 evaluators. The study therefore contributes a live, working translation tool, a reusable computational-linguistic model of Yorùbá numeral morphology, and a comprehensive evaluation framework combining automated performance analysis with human usability assessment, including an honest account of where the deployed system's output currently diverges from the formal written standard and of where its recall still trails the strongest prior benchmarks. The findings demonstrate that rule-based computational approaches remain highly effective for modelling the complex vigesimal structure of Yorùbá numerals while providing an efficient, well-received educational and linguistic resource, extending the still-developing body of Nigerian indigenous-language NLP scholarship and offering a concrete, practically usable tool in support of Yorùbá-language education and digital inclusion. Future work will focus on improving orthographic conformity with the formal written standard, integrating speech synthesis, supporting dialectal numeral variants, extending the system to ordinal numbers and currency expressions, and exploring hybrid rule-based and neural approaches for contextual numeral translation.

6. RECOMMENDATIONS

- A targeted orthographic post-processing pass should reconcile the engine's teen, subtractive-connective, and additive-connective output with the formal written-standard conventions documented by Ekundayo [8]; the recall analysis in this study shows the required corrections are systematic and localised rather than requiring a redesign of the core decomposition logic.
- Future studies should expand the usability evaluation to a larger and more geographically diverse population of Yorùbá speakers, including dialect speakers, language educators, and learners, to further validate the system across different user groups.
- Future development should integrate audio (text-to-speech) playback of generated numeral translations, drawing on existing Yorùbá speech synthesis resources, to strengthen the system's value for language learners and visually impaired users [15]; the live system's existing playback control is a natural integration point.
- The translation engine should be extended to handle the dialectal numeral variants documented for Ifẹ̀, Ìjẹ̀bú, and Ìlàjẹ Yorùbá, broadening the system's linguistic coverage beyond Standard Yorùbá.
- Subsequent work should extend the system to ordinal numbers, fractions, and currency expressions, which remain outside the scope of the present cardinal-number implementation.
- A hybrid architecture combining the present rule-based engine with a fine-tuned neural context-disambiguation layer, similar in spirit to recent context-aware numeral translation research [21], should be explored to improve handling of numerals embedded within ambiguous surrounding text.
- Institutional stakeholders, including comparable Nigerian tertiary institutions, are encouraged to incorporate openly available tools of this kind into indigenous-language curricula, in line with the broader digital-inclusion objectives in recent Yorùbá NLP policy-oriented scholarship [4].

7. ACKNOWLEDGEMENTS

The authors wish to acknowledge the Institutional Based Research (IBR) Funding of this work via the Tertiary Education Trust Fund (TETFUND).

8. REFERENCES

[1] Adetomiwa, A. 2023. Yorùbá numeral system in 21st century: Challenges and prospect. The Linguistics Association of Nigeria.

[2] Adewole, L. B., Adewole, V. T., Owoeye, M. C., Mese, F. M., & Abiola, O. B. 2026. Development of a rule-based Arabic numerals to Yorùbá translator. ABUAD International Journal of Natural and Applied Sciences, 6(1), 60-69. https://doi.org/10.53982/aijnas.2026.05-j

[3] Agbeyangi, A. O., Eludiora, S. I., & Popoola, O. A. 2016. Web-based Yorùbá numeral translation system. IAES International Journal of Artificial Intelligence, 5(4), 127-134. https://doi.org/10.11591/ijai.v5.i4.pp127-134

[4] Akomolede, K. K., & Olowojebutu, A. O. 2025. Enhancing digital inclusion through AI-based Yorùbá language localization: Challenges, solutions, and future prospects. Tech-Sphere Journal for Pure and Applied Sciences, 2(1). https://doi.org/10.5281/zenodo.15264384

[5] Aliero, R., Musa, S., & Abdulmalik, K. 2023. A review of state-of-the-art research in text normalization. International Journal of Natural Language Processing, 17(2), 55-78.

[6] Al-Saqqa, S., Sawalha, S., & Abdel-Nabi, H. 2020. Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11), 246-270. https://doi.org/10.3991/ijim.v14i11.13269

[7] Comrie, B. 2022. The arithmetic of natural language: Toward a typology of numeral systems. Macrolinguistics, 10(16), 1-35. https://doi.org/10.26478/ja2022.10.16.1

[8] Ekundayo, S. A. 1977. Vigesimal numeral derivational morphology: Yoruba grammatical competence

epitomized. Anthropological Linguistics, 19(9), 436-453. https://www.jstor.org/stable/30027551

[9] Elesemoyo, I. O. 2025. An Android-based number conversion system to Yorùbá language numeral forms. FUOYE Journal of Pure and Applied Sciences, 10(1), 102-113. https://doi.org/10.55518/fjpas.CVKU4341

[10] Elesemoyo, I. O., & Alasa, Z. A. 2025. Number sense disambiguation for Yorùbá language using multi-layered perceptron. FUOYE Journal of Engineering and Technology, 10(1), 34-39. https://doi.org/10.4314/fuoyejet.v10i1.7

[11] Elesemoyo, I. O., & Odejobi, O. A. 2022. Machine translation system for numeral in English text to Yorùbá language. Ife Journal of Information and Communication Technology, 6, 26-37.

[12] Elesemoyo, I. O., & Odejobi, O. A. 2022. Yorùbá counting system versus roman, Chinese, Hindu-Arabic, Hausa and Igbo counting systems: A computational comparison. In Current issues in descriptive linguistics and digital humanities: A festschrift in honor of Professor Eno-Abasi Essien Urua (pp. 645-654). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-2932-8_43

[13] Inuwa-Dutse, I. 2025. NaijaNLP: A survey of Nigerian low-resource languages. arXiv. https://doi.org/10.48550/arXiv.2502.19784

[14] Jimoh, E. R., Akinlolu, A. O., Akanbi, M. B., & Olojede, O. A. 2023. An android-based Yorùbá numeral translation system. FUOYE Journal of Pure and Applied Sciences, 8(3), 38-46. https://doi.org/10.55518/fjpas.PIIP2126

[15] Jimoh, T. A., De Wille, T., & Nikolov, N. S. 2025. Bridging gaps in natural language processing for Yorùbá: A systematic review of a decade of progress and prospects. arXiv. https://doi.org/10.48550/arXiv.2502.17364

[16] Molina Ríos, J., & Pedreira-Souto, N. 2019. Approach of agile methodologies in the development of web-based software. Information, 10(10), Article 314. https://doi.org/10.3390/info10100314

[17] Mustapha, R., Lawal, F. I., & Ahmad, M. A. 2023. Automated conversion of numeral to words in Hausa language. Gadau Journal of Pure and Allied Sciences, 2(2), 140-145. https://doi.org/10.54117/gjpas.v2i2.100

[18] Ninan, O. D., Iyanda, A. R., Elesemoyo, I. O., & Obasa, E. O. (2017). Computational analysis of Igbo numerals in a number-to-text conversion system. Journal of Computer and Education Research, 5(10), 241-254. https://doi.org/10.18009/jcer.325804

[19] Olakanmi, O. 2015. Yorùbá language and numerals' offline interpreter using morphological and template matching. TELKOMNIKA Indonesian Journal of Electrical Engineering, 13(1), 166-173. https://doi.org/10.11591/telkomnika.v13i1.6782

[20] Olayinka, F. D., & Eludiora, S. 2019. Development of a machine translation system for date referencing in Yorùbá language. International Journal of Computer Applications, 177(20), 32-38. https://doi.org/10.5120/ijca2019919646

[21] Sikiru, R. D., Shogbamu, O., Soronnadi, A., Adekanmbi, O., & Adebara, I. 2025. Context-aware neural machine translation of English numerical expressions to Yorùbá: A fine-tuning approach for financial and general domains [Conference session]. Women in Machine Learning Workshop, NeurIPS 2025. https://openreview.net/forum?id=DMinxnkT7P

[22] Tang, W., Yu, J., Li, Y., Zhao, Y., Zhang, W., Feng, W., Zhang, M., & Yang, H. (025. Investigating numerical translation with large language models. In ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1-5). IEEE. https://doi.org/10.1109/ICASSP49660.2025.10887726

[23] Xu, Y., Liu, E., & Regier, T. 2020. Numeral systems across languages support efficient communication: From approximate numerosity to recursion. Open Mind, 4, 57-70. https://doi.org/10.1162/opmi_a_00034