IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 5, No. 4, December 2016, pp. 127~134
ISSN: 2252-8938

Web-Based Yorùbá Numeral Translation System

Agbeyangi A. O.[1], Eludiora S. I.[2], Popoola O. A.[2]
[1]Department of Computer Engineering, Moshood Abiola Polytechnic, Abeokuta, Ogun State-Nigeria
[2]Department of Computer Science & Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

Article Info

Article history:

Received Aug 12, 2016
Revised Oct 16, 2016
Accepted Nov 19, 2016

Keyword:

Machine translation
Number
Yorùbá numeral

ABSTRACT

Yorùbá numerals have been seen as one of the most interesting but quite complicated numeral system. In this paper we present the development of a web-based English to Yorùbá numeral translation system. The system translates English numbers both in figure and text to its standard Yorùbá form. The computational processes underlying both numerals were used to formulate the model for the work. Unified Modeling language (UML) and Automata theory was used for the system design and specification. The designed system was implemented using Google Web App Engine with support for python. The result of the system evaluation using mean opinion score approach shows that the system gives a recall of 100% on all the output considered.

Corresponding Author:

Agbeyangi A. O.,
Department of Computer Engineering,
Moshood Abiola Polytechnic,
Abeokuta, Ogun State-Nigeria.
Email: abayomiagbeyangi@gmail.com

1. INTRODUCTION

According to [1], number is an arithmetical value, expressed by a word, symbol, or figure, representing a particular quantity and used in counting and making calculations. Natural numbers are the familiar positive whole numbers 1, 2, 3, etc., and they clearly play an essential part in many mathematical activities; for example, counting and arithmetic. In addition to their practical role, numbers also have a central place in mathematical theory [2]. Also in [3], number is explained as an abstract idea of collection of things but numerals are man-made symbols that represent the numbers. Numbers are always the same value, no matter what symbol or word is used to represent them. For example, 1, I, i are numerals that represent the same number as one. In [4], the numeral system or system of numeration is seen as a mathematical notation for representing numbers of a given set, using digits or other symbols in a consistent manner.

Numeral in whatever language and in whichever forms/patterns it takes is a clear manifestation of the ability of man to manipulate his linguistic resources to cater for his communicative needs. Numerals appear an aspect of language study which researchers, scholars and linguists pay less attention to. This neglect could be connected to certain impression that there is little or nothing else to say about it. Part of our linguistic knowledge is contained in the ability to express our ideas in our mother tongues. Our linguistic knowledge remains insufficient if we are unable to count fluently, analyze how the counting is done, and understand the manipulations therein [5]. As noted in [6], counting and or numbering is an integral and inseparable part of the grammar of any language because there is hardly any meaningful linguistic discourse in a language that does not make reference to quantity, size, time, distance and weight in definite numbers. Also, the use of numbers and their power in capturing concepts have makes them indispensable in effective communication [7]. In the society today, numbers are seen as an indispensable tool in our day-to-day activities. [8] stated that, key advancement in civilization can be traced to the conception, invention, representation, and manipulation of numbers to facilitate accurate rendering of measurable objects. This has made the use of numbers an important tool within the society as they are mostly used in trade, cosmology, mathematics, divination, music, medicine, etc.

The Hindu-Arabic numeral (as shown in Figure 1) has been seen as a form of English numeral since most of the numbers used in English numeral were borrowed from other language particularly the Hindu-Arabic which is still the most commonly used symbolic representation of numbers due to its simplicity and the fact that it requires little memorization to represent practically any number [9]. In an early study of the Yorùbá numeral system, [10] shows how large numbers could be represented as an arithmetic combination of the basic number units and reveals that the subtraction operation plays an important role in its number naming. The peculiarity in the Yorùbá numerals was that: very different is the framework of the Yorùbá, it can boast of a greater number of radical names of numerals, and to a large extent makes use of subtraction [10]. Arguably, English numeral uses the decimal (base 10) form of numeral notation while Yorùbá numeral uses vigesimal (based 20) extensively.

Hindu-Arabic	0	1	2	3	4	5	6	7	8	9
Western Arabic-Indic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Eastern Arabic-Indic (Persian and Urdu)	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
Devanagari (Hindi)	०	१	२	३	४	५	६	७	८	९

Figure 1. Hindu-Arabic Numerals (source: ([15]))

Yorùbá is a major ethnic group in Nigeria. Presently, they are found in large concentration along the West African coast as well as other major cities of the world. Their communities in the Diaspora can be found in Brazil, the Caribbean: Trinidad and Tobago, Jamaica, Europe and the United States of America [11].

The rest of the paper is organized as follows: section 2 examines related works; section 3 gives the research methodology; section 4 discusses the results, while section 5 concludes the paper.

2. RELATED WORKS

In [12], Yorùbá numeral was explain as a form of a vigesimal numeral system i.e. its numbers are mostly represented in base 20. But in [13], it was argued that where the European decimal system is based upon units of ten and the functions of additions and multiplication as it is used in English numeral, Yorùbá numeral system is based upon units of both ten and twenty and the function of addition, multiplication and subtraction. [14] Further argued that, Yorùbá numeral system is not fully vigesimal, and that elements of decimal and quinary (base 5) are also used in its numeral representation.

In the study by [12] on vigesimal numeral derivational morphology, it was stated that numerals are innumerably infinite i.e. there is no longest numeral and that numerals can be created or derived especially numerals of higher values with low usage. He further stated that, in Yorùbá numeral system, multiple representations exists for those numerals which have to be created or derived because their functional load is low. He then classified Yorùbá numerals into sixteen (16) basic forms which are: One (1) to ten (10), Twenty (20), thirty (30), two hundred (200), three hundred (300), four hundred (400) and twenty thousand (20000). It was further identified that some various subclasses of numerals exist for Yorùbá numbers and they are given as: the noun form (those which can operate like nouns), cardinal form (e.g méta, mérin, etc.) and ordinal form (e.g. èkíní, èkejì, èketa etc.).

In the study conducted by [14], it was explain that, apart from Yorùbá, other African languages with vigesimal numeral system are; Madingo, Mundo, Logone, Nupe, Nembe, Bingo, Efik, Vie, Igbo and Afadeh. It was emphasize that Yorùbá counting system has lexemes for basic numbers from one (1) to ten (10) and six higher numerals (twenty (20), thirty (30), two hundred (200), three hundred (300), four hundred (400), and twenty thousand (20000). It was also stated that another set of basic numerals used in Yorùbá numeral are *okòó* (20), *òjì* (40), *òta* (60) and *ọ̀rin* (80) as they are used in multiples of 100 and between 200 and 20000. Okòó is used when *ólé* (addition) or *ódín* (subtraction) is added to the vigesimal ogún and used with the multiplication formation (e.g okòólérúgba - 220). The study also examines numeral operators used in the derivation of Yorùbá numeral as *lé* for addition, *dín* for subtraction and óná for multiplication. It was further noted in the study that, in Yorùbá numeral system, when *ẹẹdín* is used with a number, it implies that the number must be reduced by a certain value.

The study by [16], shows that Yorùbá numeral maintains a VCV (disyllabic) open syllabic structure while English numeral has a monosyllabic closed syllabic structure CVC for numeral 1, 5, 9 and 10, CVCC for 6, and VC for 8. The study also agreed with [14] and [12] that, numerals 200, 300 and 400 are basic in Yorùbá numeral system while numerals 500, 700 and 900 are derived through a combination of subtraction and multiplication whereas 600, 800 and 1000 are derived by multiplying 200 by 3, 4, and 5 respectively. The work further explain that, in Yorùbá *'l'* and *'n'* are allophones, therefore, while *'n'* goes with nasalised vowel, *'l'* goes with oral vowels. Such that in *dín-ní-ogún* (less than twenty), the combination of *ní-ogún* becomes *lógún* because the nasal vowel *'l'* after *'n'* is deleted at the junction.

The pattern for the generation of Yoruba numerals was highlighted in [17] as follows: One to ten are basic words and eleven to fourteen is expressed as 1 + 10, 2 + 10, 3 + 10 and 4 + 10 respectively. Fifteen to nineteen are expressed as 20 - 5, 20 - 4, 20 - 3, 20 - 2, 20 - 1 and twenty "ogun" is a basic word. Twenty-one to twenty-four are expressed as 20 + 1, 20 + 2, 20 + 3 and 20 + 4. Twenty-fve to twenty-nine are expressed as 30 - 5, 30 - 4, 30 - 3, 30 - 2 and 30 - 1. Thirty "ọgbọ̀n" is another basic word. A pattern similar to the one above is followed for thirty-one to thirty-four and thirty-five to thirty-nine. Forty "ogójì" from ogún + èjì is expressed as 20 x 2. The pattern of the addition of 41 to 44 and subtraction for 45 to 49 is followed for numbers after fifty, sixty, seventy, eighty, etc. Fifty, àdọ̀ta, is 60 - 10. This pattern is followed for àádóòrin 70 (80 - 10), àádóòrùn-ún 90 (100 - 10), àádóòfà 110 (120 - 10), àádóje 130 (140 - 10), àádóòjoò 150 (160 - 10), àádóòsàn-án 170 (180 - 10), and àádóòwàá 190 which is also expressed as igba-odín-mẹ́wàá, or mẹ́wàá-dín-nígba (200 - 10).The pattern of multiplication used for forty is followed for 60 ọ̀gọ́ọ̀ta 20 x 3, 80 ọ̀gọ́ọrin 20 x 4, 100 ọ̀gọ́ọrùn-ún 20 x 5, 120 ọ̀gọ́ọ̀fà 20 x 6, 140 ogóje 20 x 7, 160 ọgọ́ọ̀jọ 20 x 8, 180 ọ̀gọ́ọ̀sàn-án 20 x 9 and another basic word, igba is used for 200. Therefore, the basic Yorùbá numerals are; one to ten (óókan - ẹ́ẹ̀wá). Twenty (ogún), thirty (ọgbọ̀n), two-hundred (igba), three-hundred (ọ̀dúnrún), four-hundred (irínwó) and twenty-thousand (ọ̀kẹ́ Kan) [12]. All other numerals apart from the basic numerals are derived using rules. These rules will the discussed extensively under methodology.

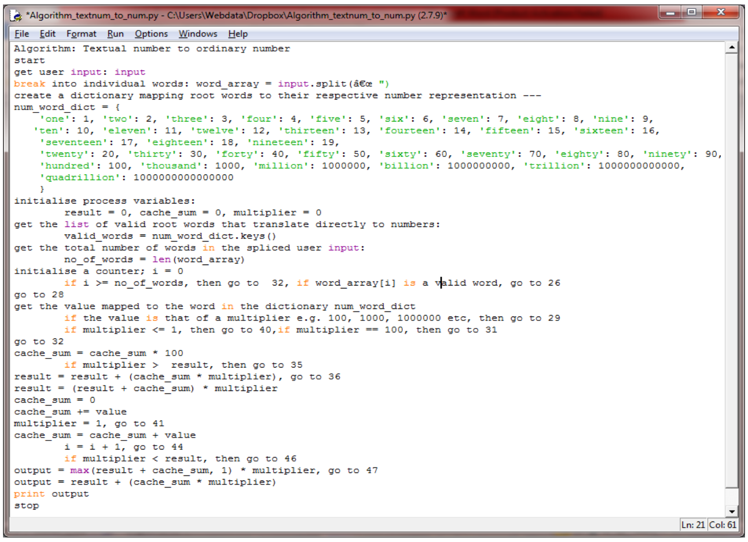
3. METHODOLOGY

The system is designed as an online web application for easy access to users. The pattern for the Yorùbá numeral system was used to develop the re-writes rule for the Translation process.
The re-write rules are given below:

S → NUM
NUM → NP | NUM SN
SN → VV NP
NP → DIGIT | MP | VP NP
NP → REDUCE MP | NP M
MP → M | MP NP
VP → DIGIT V

The production rules given above capture the various segments in the translation process. In the production rule, NP, represent noun phrase or simply number phrase, MP represents multiplicative base, and NUM is the generated number. As reported in [14], Yorùbá numeral system is very methodical, therefore, an efficient computational system is necessary to accurately derived it numeral representation. It is as seen as a near difficult form of numeral system. There are five important processes in the model. First, is the part that converts the textual number to numbers in figure (the algorithm used for this is shown in Figure 2), second stage is the number decomposition process, where the numbers are expressed as a sum of smaller numbers in harmony with their sub-grouping. The output of this process is the magnitude stack. Next, there is a process that generates the possible forms of a single number. This is done by careful combinations of the neighbouring elements of the magnitude stack and parsing them with the designed numeral grammar. The third process is where tokens of the number form are converted to their equivalent lexical forms, and the final process is where the morphophonological rules employed in numbers naming.

The generated production rule was simulated using Automata. Its behaviour was expressed as an abstract machine. JFLAP, a Java-based Automata simulation tool was used. The CFG for the Yorùbá numeral translation system is as shown in Figure 3. The FSA has six states (q0 to q5). The initial state is q0 while the final state where the state transitions are accepted or rejected is q3. In Figure 4, a rule is simulated to show if the automaton accept the input or not.



```
Algorithm: Textual number to ordinary number
start
get user input: input
break into individual words: word_array = input.split(â€œ ")
create a dictionary mapping root words to their respective number representation ---
num_word_dict = {
    'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7, 'eight': 8, 'nine': 9,
   'ten': 10, 'eleven': 11, 'twelve': 12, 'thirteen': 13, 'fourteen': 14, 'fifteen': 15, 'sixteen': 16,
    'seventeen': 17, 'eighteen': 18, 'nineteen': 19,
    'twenty': 20, 'thirty': 30, 'forty': 40, 'fifty': 50, 'sixty': 60, 'seventy': 70, 'eighty': 80, 'ninety': 90,
    'hundred': 100, 'thousand': 1000, 'million': 1000000, 'billion': 1000000000, 'trillion': 1000000000000,
    'quadrillion': 1000000000000000
    }
initialise process variables:
        result = 0, cache_sum = 0, multiplier = 0
get the list of valid root words that translate directly to numbers:
        valid_words = num_word_dict.keys()
get the total number of words in the spliced user input:
        no_of_words = len(word_array)
initialise a counter; i = 0
        if i >= no_of_words, then go to  32, if word_array[i] is a valid word, go to 26
go to 28
get the value mapped to the word in the dictionary num_word_dict
        if the value is that of a multiplier e.g. 100, 1000, 1000000 etc, then go to 29
        if multiplier <= 1, then go to 40,if multiplier == 100, then go to 31
go to 32
cache_sum = cache_sum * 100
        if multiplier >  result, then go to 35
result = result + (cache_sum * multiplier), go to 36
result = (result + cache_sum) * multiplier
cache_sum = 0
cache_sum += value
multiplier = 1, go to 41
cache_sum = cache_sum + value
        i = i + 1, go to 44
        if multiplier < result, then go to 46
output = max(result + cache_sum, 1) * multiplier, go to 47
output = result + (cache_sum * multiplier)
print output
stop
```

Figure 2. Algorithm to Convert Textual Number to Number

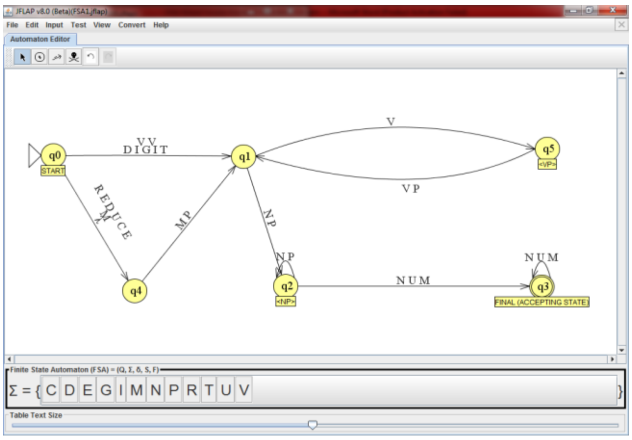


Figure 3. Re-Write Rule Automaton

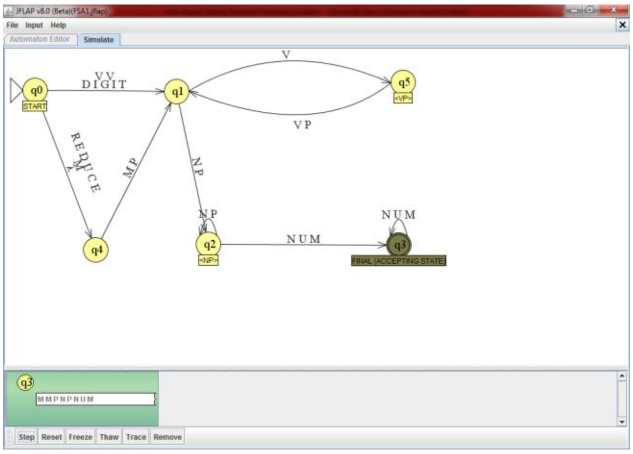


Figure 4. Re-write Rule Automaton Simulation

Figure 5 also show the production rule simulation with multiple inputs where accepted and rejected input combination are expressed.

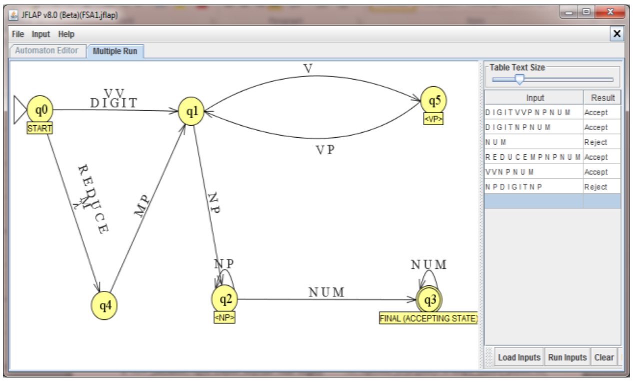


Figure 5. Re-write Rule Automaton Simulation with Accepted and Rejected Inputs.

3.1. Data Collection

The various data used for the research are as follow:

1. The set of lexemes also known as the basic number in Yorùbá numeral system, which are: DIGIT = {okan (1), eji (2), eta (3), erin (4), arun-un (5), efa (6), eje (7), ejo (8), esan-an (9), ewa (10), ogbon (30), odunrun (300), irinwo (400)}
2. The sets of multiplicative base; M = {ogun (20), igba (200), oke (20000)}
3. The set of lexical affixes showing arithmetic operation; V = {le ni (+), din ni (-)}
4. The set of operator between phrases, VV = {o le (++), o din (--)}
5. The set of implied subtraction operators; REDUCE = {aadin (reduction by 10), eedin (reduction by 5, 100 and 1000).

Every other entry in the database (stack) is generated by the rules.

3.2. Development Tools

The system implementation was done using the following software tools:

1. Google App Engine: Is the platform for developing and hosting the application. The platform is a Google-managed data centers for developing and hosting web application.
2. Python: This is the main programming tool used. Its support for web programming provides the flexibility used for the application development.
3. NLTK: It was used for parsing and generating the parse tree.
4. JFLAP: It was used in simulating the production rule used.

The system use case diagram is shown in Figure 6. It defines the overview of the usage requirement of the system as described in the research work.

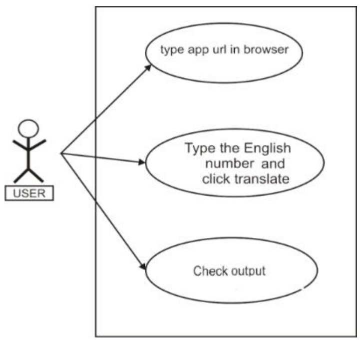


Figure 6. System Use Case Diagram

4. RESULT AND DISCUSSIONS

The application can be access at https://onka-yoruba.appspot.com. The screenshots of the web application is as shown in Figure 7 to Figure 10. In Figure 7, forty-five in text was translated to “márùn-úndínlààdọ́ta” in Yorùbá. In Figure 8, the save forty five in figure is converted to same “márùn-úndínlààdọ́ta” Yorùbá. The power of the system to translate higher number was tested in Figure 9 and 10 where one million six hundred and twenty in text and figure was translated to “ààdọ́ta ọ̀kẹ́ ó lé okòólélẹgbẹ̀ta” in Yorùbá.

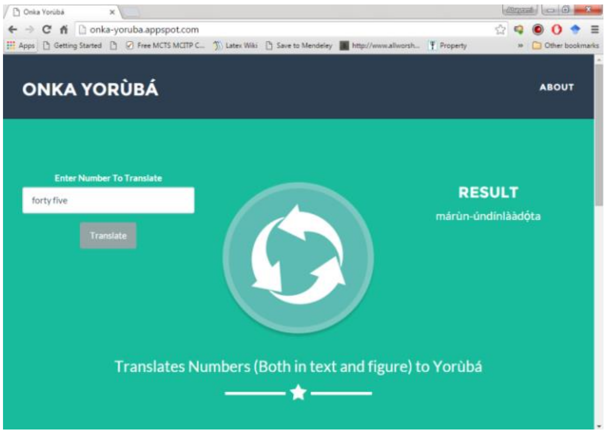


Figure 7. Application Output Sample I

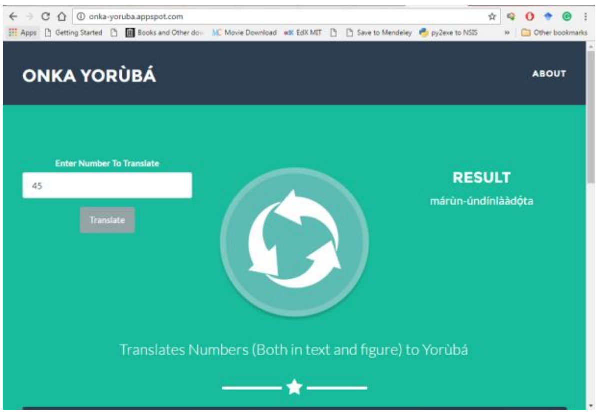


Figure 8. Application Output Sample II

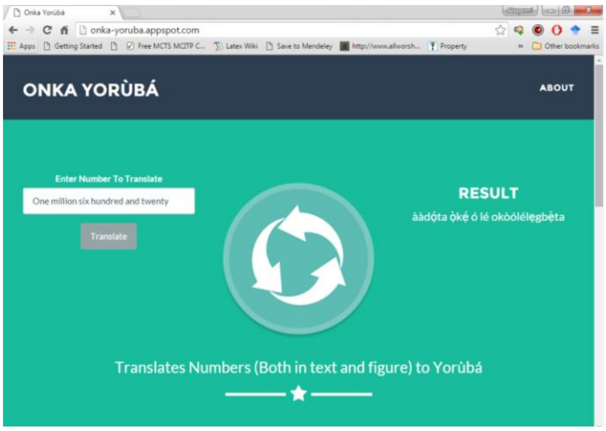


Figure 9. Application Output Sample III

Figure 10. Application Output Sample IV

5. CONCLUSION

The development of the system, though tasking but entails an interesting area of research due to the fact that the numeral system of Yorùbá language is becoming obsolete as its use among the young speakers of the language is diminishing. This study has developed a useful artifact and algorithm which will be of important to anyone interested in Yorùbá numeral system. Other area for further research includes the development of translation model for Yoruba dates, ordinal numbers and the translation from Yoruba to English numeral.

REFERENCES

[1] Hurford, JR. The linguistic theory of numerals, volume 16. Cambridge University Press. 1975.
[2] Rips, LJ, Bloomfield, A, Asmuth, J. From numerical concepts to concepts of number. Behavioral and Brain Sciences, 2008; 31(6): 623-642.
[3] Chowdhury, K, Baishya, A A Comparative Study of Early Egyptian, Babylonian and Mayan Number System. *International Journal of Mathematical Archive (IJMA).* 2011.
[4] Ifrah, G. The universal history of numbers: From prehistory to the invention of the computer (D. Bellos, I. Monk, EF Harding, & S. Wood, Trans.). 1999.
[5] Babarinde, O. The Linguistic Analysis of the Structure of the Yoruba Numerals. *Journal of Literature, Languages and Linguistics*, 2013; 1:78-88.
[6] Omachonu, GS. Derivational Processes in Igala Numeral System. *Journal of universal language*. 2011; 12(2):81-101.
[7] Goyvaerts, D. Counting in Logo. *Anthropological Linguistics*, 1980: 317-328.
[8] Zaslavsky, C. Africa Counts and ethnomathematics. For the Learning of Mathematics, 1994: 3-8.
[9] Bailey, DH, Borwein, JM. High-precision numerical integration: Progress and challenges. *Journal of Symbolic Computation.* 2011; 46(7): 741-754.
[10] Mann, A. Notes on the Numeral System of the Yoruba Nation. *Journal of the Anthropological Institute of Great Britain and Ireland*. 1887: 59-64.
[11] Atanda, JA. The Yorùbá people: their origin, culture and civilization. The Yorùbá: History, Culture and Language. 1996: 3-34.
[12] Ekundayo, S. Vigesimal numeral derivational morphology: Yoruba grammatical competence epitomized. Anthropological Linguistics. 1977: 436-453.
[13] Hall, G. Yoruba Numbers'. In Folklore Forum, 1973: 1.
[14] Akinade, OO, Odejobi, OA. Computational modelling of Yoruba numerals in a number-to-text conversion system. *Journal of Language Modelling*. 2014; 2(1):167-211.
[15] Smith, DE, Karpinski, LC. The Hindu-Arabic Numerals. Courier Corporation. 2013.
[16] Babarinde, O. The Linguistic Analysis of the Structure of the Yoruba Numerals. *Journal of Literature, Languages and Linguistics*. 2013; 1:78-88.
[17] Oyetade, BA. Comparative Number Words of Five Àkókò Dialects and Ẹ̀dó with Standard Yorùbá. *Journal of Nigerian Languages and Literatures*. 1996; 2:17-46.
[18] Atanda, JA. An Introduction to Yorùbá History. Ibadan University Press. 1980.

[19] Awobuluyi, O. Towards a typology of coalescence. *Journal of West African Languages*. 1987; 17(2): 5-22.
[20] Bird, S. *NLTK: the Natural Language Toolkit.* In Proceedings of the COLING/ACL on Interactive presentation sessions. 2006: 69-72. Association for Computational Linguistics.
[21] Bird, S, Klein, E, Loper, E. Natural language processing with Python. O'Reilly Media, Inc. 2009.
[22] Ifrah, G. From one to zero: A universal history of numbers. New York: Penguin Books. 1985.