

Research Article

Computational Analysis of Igbo Numerals in a Number-to-text Conversion System

Olufemi Deborah NINAN¹ , Abimbola Rhoda IYANDA^{2*} , Isaac Olufemi ELESEMOYO³ , Esther Olamide OBASA⁴ 

¹ Lecturer, Computer Science and Engineering Department Obafemi Awolowo University, deborah.ninan@gmail.com

² Lecturer, Computer Science and Engineering Department Obafemi Awolowo University, abiyanda14@gmail.com

³ PG Student, Computer Science and Engineering Department Obafemi Awolowo University, ielesemoyo@gmail.com

⁴ PG Student, Computer Science and Engineering Department Obafemi Awolowo University, obasaolamide@gmail.com

* Corresponding Author: abiyanda14@gmail.com

Article Info

Received: July 03, 2017

Accepted: November 17, 2017

Online: December 03, 2017

Keywords: syntax-based, Igbo language, translator, numeral, computational.

Abstract

System for converting Arabic numerals to their textual equivalence is an important tool in Natural Language processing (NLP) especially in high-level speech processing and machine translation. Such system is scarcely available for most African languages including the Igbo language. This translation system is essential as Igbo language is one of the three major Nigerian languages feared to be among the endangered African languages. The system was designed using sequence as well as activity diagram and implemented using the python programming language and PyQt. The qualitative evaluation was done by administering questionnaires to selected native Igbo speakers and experts to provide preferred representation of some random numbers. The responses were compared with the output of the system. The result of the qualitative evaluation showed that the system was able to generate correct and accurate representations for Arabic numbers between 1-1000 in Igbo language being the scope of this study. The resulting system can serve as an effective teaching and learning tool of the Igbo language.



To cite this article: 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>

Introduction

Numbers have been used for counting and measuring for over 5,000 years by the ancient culture for effective communication in education, trading and agriculture (Eves, 1969). Nigeria is a multilingual country with over five hundred languages spoken by different ethnic groups across her regions however, the three major indigenous languages are, *Yoruba*, *Igbo* and *Hausa*. The languages spoken in Nigeria are not evenly distributed, for instance in the South-West part of Nigeria, *Yoruba* is largely spoken; *Igbo* is largely spoken in

the South-East part of Nigeria; while in the North-West part of Nigeria, *Hausa* is largely spoken (Yusuf, 2007).

In the past, some cultural, religious, and socio-political organization of the *Igbos* had their various re-served number symbols. Following the extensive and continued contact with the west and the worldwide desire to go metric and decimal, the *Igbos* in the recent past decimalized their number system (Ogomaka, 2005). In essence, there are the traditional (as reconstructed in Table 1) and the decimalized *Igbo* number systems.

Table 1. Traditional and Decimalized *Igbo* Number representation of Hindu-Arabic Numerals

Hindu-Arabic Numerals	Traditional <i>Igbo</i> Numbers	Decimalized <i>Igbo</i> Numbers
1	<i>Otu</i>	<i>Otu</i>
2	<i>Abuo</i>	<i>Abuo</i>
3	<i>Ato</i>	<i>Ato</i>
4	<i>Ano</i>	<i>Ano</i>
5	<i>Ise</i>	<i>Ise</i>
6	<i>Isii</i>	<i>Isii</i>
7	<i>Asaa</i>	<i>Asaa</i>
8	<i>Asato</i>	<i>Asato</i>
9	<i>Itolu</i>	<i>Itolu</i>
10	<i>Iri</i>	<i>Iri</i>
11	<i>Iri na otu</i>	<i>Iri na otu</i>
12	<i>Iri na abuo</i>	<i>Iri na abuo</i>
13	<i>Iri na ato</i>	<i>Iri na abuo</i>
14	<i>Iri na ano</i>	<i>Iri na ano</i>
15	<i>Iri na ise</i>	<i>Iri na ise</i>
16	<i>Iri na isii</i>	<i>Iri na isii</i>
17	<i>Iri na asaa</i>	<i>Iri na asaa</i>
18	<i>Iri na asato</i>	<i>Iri na asato</i>
19	<i>Iri na Itolu</i>	<i>Iri na Itolu</i>
20	<i>Ogu</i>	<i>Iri Abuo</i>
21	<i>Ogu na otu</i>	<i>Iri Abuo na otu</i>
22	<i>Ogu na Abuo</i>	<i>Iri Abuo na Abuo</i>
27	<i>Ogu na asaa</i>	<i>Iri Abuo na asaa</i>
30	<i>Ogu na iri</i>	<i>Iri ato</i>
40	<i>Ogu Abuo</i>	<i>Iri ano</i>
97	<i>Ogu ano na iri na asaa</i>	<i>Iri itolu na asaa</i>
100	<i>Ogu ise</i>	<i>Nari</i>
154	<i>Ogu asaa na iri na ano</i>	<i>Otu Nari na iri ise na ano</i>
176	<i>Ogu asato na iri na isii</i>	<i>Otu Nari na iri</i>
200	<i>Ogu iri</i>	<i>Nari abuo</i>
335	<i>Ogu iri na isii na iri na ise</i>	<i>Nari ato na iri ato na ise</i>
400	<i>Nnu</i>	<i>Nari ano</i>
900	<i>Nnu abuo na Ogu ise</i>	<i>Nari Itolu</i>
1000	<i>Nnu abuo na ogu iri</i>	<i>Otu Puku</i>

Companies use machine translation software to translate manuals and track revisions, governments also use them to translate web pages and other net traffics (Knight, 1997). Hutchins (2005) reported that there is a huge number of translation systems covering many

but majorly European languages (English, French, German, Italian, Portuguese, Russian, and Spanish). However, very few systems are available for African languages.

Software for converting numbers to their textual equivalence is an important tool in Natural Language Processing (NLP) especially in high-level speech processing and machine translation. Such software was not available for most African languages until recently; Akinade (2014) developed a computational model for Yoruba numerals. However, until now there is no reported software for the *Igbo* language numerals. *Igbo* (*Asusu Igbo*) being one of the three major Languages and also an interesting language in Nigeria, has a complex numeral system which makes people draw back from learning the language.

The underlying thesis of this work is to explore how *Igbo* numbers (*OnuOgugu Igbo*) are represented and to develop a system that automatically converts cardinal numbers into their corresponding textual representations.

Review of Literature

Knight and Graehl (1998) have reported that translation is more complicated for language pairs that employ very different alphabets and sound systems. It was observed that the determinant of development of machine translation include the availability of new technology, political, social and economic need for change. However, despite the advances, machine translation still represents only a tiny percentage of the numerous pages translated (Craciunescu *et. al*, 2004). The primary advantages of machine over human translation are speed, cost, availability, and consistency and web- and PC-based machine translation will continue to have an important role in the critical and growing natural language translation market (Ablanedo *et. al* 2007). Estimate by Craciunescu *et. al* (2004) stated that to produce a good translation of a difficult text, a translator cannot process more than 46 pages or 2,000 words per day.

According to Hammarstrom (2009), a numeral system is a spoken normed expression used to denote the exact number of objects for an open class of objects in an open class of social situations with the whole speech community in question. This definition explains that numeral systems are focused on the written form of spoken numbers with acceptable semantics within the majority of a language group. In addition, numeral system should be applicable to a wide range of applications such as counting and monetary transaction (Akinade, 2014).

As a system, each has a set of rules, principles and properties that govern the formulation and existence of number names and numerals other than the basic ones (Ogomaka, 2005). Many cultures (as regards distinct language groups) formulated and developed to some degree distinct number systems but the majority of such systems did not provide the concept of number and quantification. Dixon (2011) in his study of Australian languages highlighted the lack of system of numbers within the Australian vocabularies. Everett (2005) also shows the absence of a concept of counting within the grammar and culture of the Pihara. Akinade (2014) opines that despite the fact that numbers are represented differently across languages employing different computational techniques with varying complexities, numeral systems still share some common properties.

Numeral systems have evolved over the years in search of more effective ways to count and represent numbers. The body/finger and object counting is probably the earliest form of number system. The body/finger counting is common in Africa especially in Rwanda and *Tanzania* where they form an integral part of all transaction in the market (Zaslavsky, 1973) and body parts are touched in sequence to communicate numbers.

The traditional Igbo number system

The traditional *Igbo* number system is no exception to these provisos. In everyday use, numbers have both cardinal and ordinal sense. In the cardinal sense, the traditional *Igbos* used numbers; as quantitative adjectives, or in answering the question how many. Thus people talk of *ego iri, ugu ego ugu iri na ise*, etc. an ordinal number indicates not only how many but also answers the question in what order. For example, the day of the month is ordinal (Wilder, 1973). Traditional *Igbo* people, like other traditional people used and interpreted numbers in a number of ways. In very broad terms, the traditional *Igbo* people had: (i) the everyday or commonplace use and (ii) the humanistic (religious, mystic, socio-political) uses and interpretation of numbers (Ogomaka, 2005).

The Traditional *Igbo* Number System (TINS) is vigesimal (a base twenty system). It also has a minor base or a sub-base which is decimal or denary (a base ten system). Counting in the system is done in bundles of twenty or possible positive integral powers of twenty. Consequently most of the numbers that are positive integral powers of twenty within the range of counting of the traditional *Igbo* person, have distinct names. The developers of the traditional *Igbo* number system were most probably informed or guided by the number of

fingers and toes a normal person has, in deciding the main base (vigesimal) and minor base (decimal) of the system. However, Mbah *et. al* (2014) concluded that speakers of *Igbo* language rarely make use of the traditional numbering and counting systems because British decimal system of numbering has, overshadowed vigesimal systems.

There are several computational processes employed in deriving higher numerals in different languages. For example, using the multiplicative principle in *Igbo* translation, it is only the main base that is multiplied by integers. The use of the subtractive principle involves usually subtracting the number one, two, three, four or five (though rarely) from the main base, the integral multiple or power of the main base or the minor base.

It has been observed that the regularity of numeral systems is strongly connected to the concept of base. In numeral systems, the number 'n' is a base of a numeral system if the next higher base (or the end of the normed expressions) is a multiple of 'n'; and majority of the expressions for numerals between n and the next higher base are formed by (a single) addition or subtraction of 'n' or a multiple of 'n' in conjunction with expressions for numbers smaller than 'n' (Hammarstrom, 2009).

The numeration system of the *Igbos* is at present decimal/denary. The Hindu - Arabic numerals have been adopted which justifies the semblance between decimalized *Igbo* numeration system and the English number system. Numbers 1-10 which forms the basic morphemes within the *Igbo* numeral system consist of digits which are not broken down into smaller units. The structures of these numbers follow a unary tree. Representation of numerals from 11-19 requires the addition of a DIGIT to 10 such as *iri na otu* ($11 = 10 + 1$), *iri na asato* ($18 = 10 + 8$), *iri na itolu* ($19 = 10 + 9$). The *Igbo* lexical representation of numbers 11 and 736 are formed as an additive concatenation of the terms for numbers 1 and 10 as shown in Figure 1a and b respectively but the number 20 uses multiplicative concatenation as shown in Figure 1c.

The computational process involved in *Igbo* numerals becomes remarkably obvious after the decimal number 19. The process henceforth does not follow the additive principle which involves the addition of 'na' between two *Igbo* basic number. While the numeral 20 uses a multiplicative principle i.e. the direct association of two basic *Igbo* numerals, the number 125 involves both the additive and multiplicative principle as shown in Figure 1d.

Model description

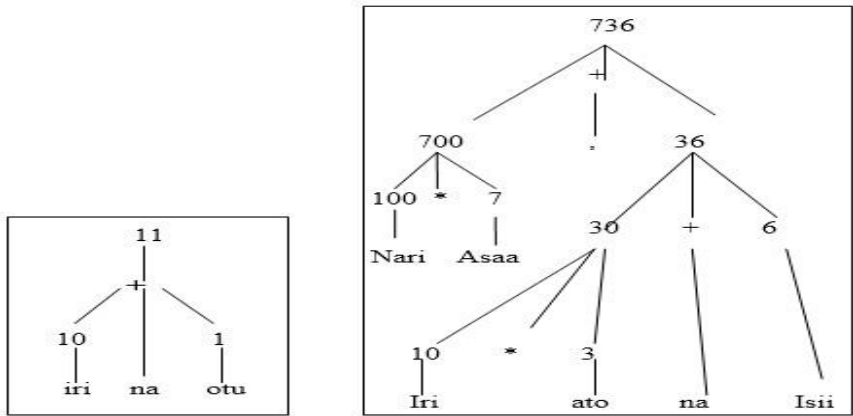
The activity flow of the translation processes is presented in Figure 2. In the entire model, the input is a number and the output is the *Igbo* text equivalence of the input. The steps required to achieve the process of converting number to *Igbo* text are discussed in the following subsections:

Tokenization and number identification

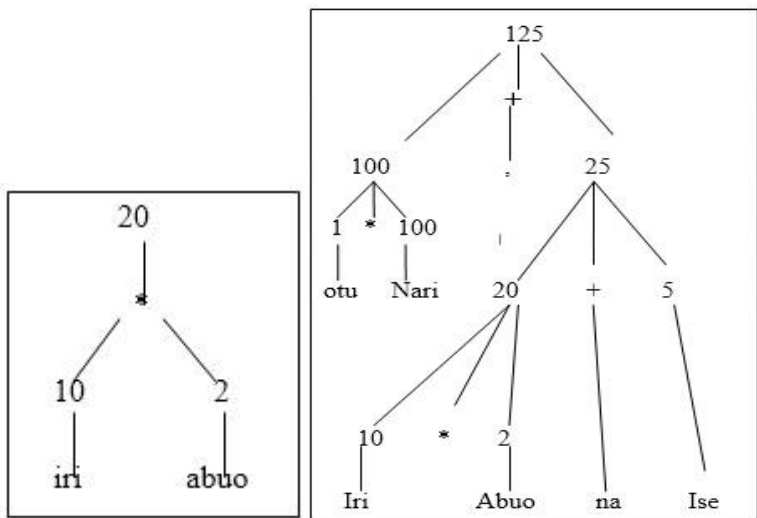
This process is performed when the input is a text file. Tokenization takes the input text and breaks it into sentences marked by a new-line, and each sentence is further broken-down into chunks called tokens. Tokenization was done using white spaces as delimiters. This process was achieved using the *split()* function of the String class. This function takes a String and separates them using white space. The output of the *split()* function is a List object containing all the tokens. Each token was tested with a set of hand-crafted rules to identify the number type. The number type considered is nominal numbers. The set of rules used were formulated using regular expressions. For example: $\backslash d+[\backslash d]^*$ e.g. 1,2,13,26,273,1000 etc.

Number decomposition

This stage represents a number as a sum of smaller numbers, which are best, handled as a number phrase. The first process is to generate the magnitude stack from the given number. The magnitude stack contains numbers, which can be derived as multiples of multiplicative bases (i.e.1, 10, 100, and 1000). The number decomposition process generates four new numbers (d_0, d_1, d_2, d_3) from the given number. Therefore, it can be stated that: $\text{Number} = d_3 + d_2 + d_1 + d_0$ such that d_0, d_1, d_2 and d_3 can be expressed using just the basic lexical items of the *Igbo* numeral system. Table 2 shows the Algorithm for the magnitude generator. The magnitude stack of some numbers generated using the Algorithm is presented in Table 3. Any of d_3, d_2, d_1 and d_0 that is equal to zero was completely removed from the magnitude stack. The following rules hold for the content of the magnitude stack:



(a) Parse Tree for number 11. (b) Parse Tree for number 736.



(c) Parse Tree for number 20 (d) Parse Tree for number 125

Figure 1. Tree structure of *Igbo* composite numbers

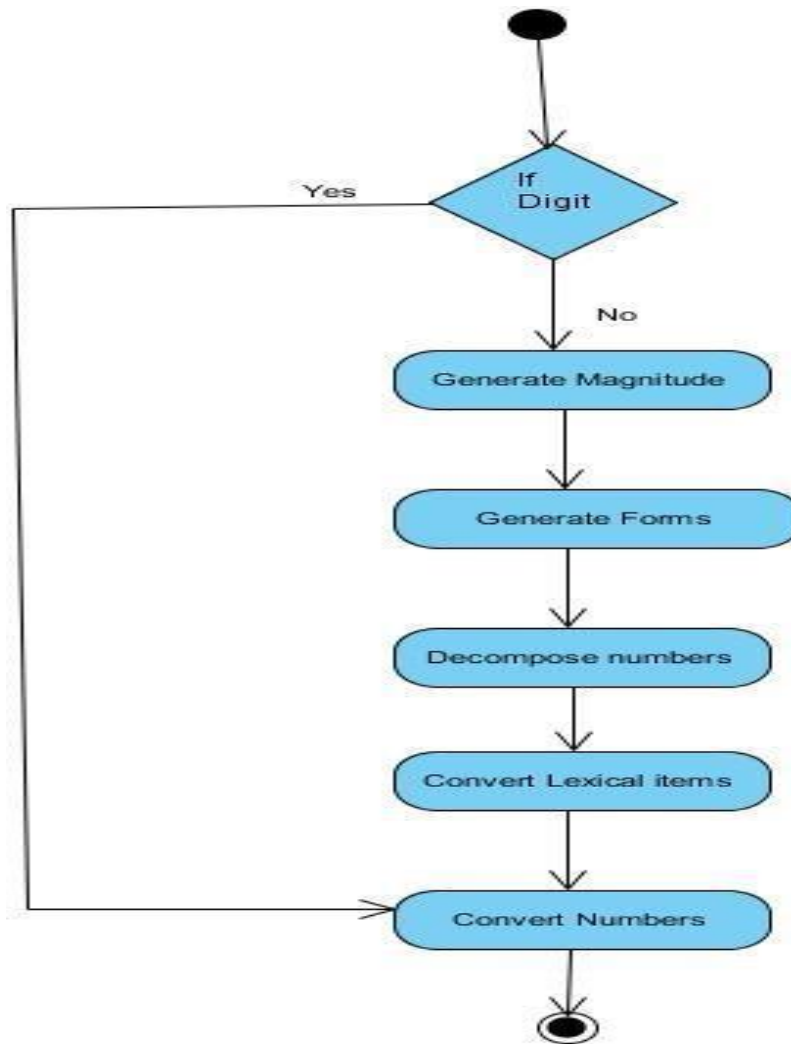


Figure 2. Activity Flow

- i. d_3 is a multiple of 1000 (*Puku*). ($d_3 = *n \mid \leq n \leq$) $n \in \text{DIGIT}$
- ii. d_2 is a multiple of 100 (*Nari*). ($d_2 = *n \mid \leq n \leq$) $n \in \text{DIGIT}$
- iii. d_1 is a multiple of 10 (*Iri*). ($d_1 = *n \mid \leq n \leq$) $n \in \text{DIGIT}$
- iv. d_0 takes values from 0 to 9 i.e. $d_0 \in \text{DIGIT}$

For example, the magnitude stack generated for number 1,234 was $[d_3, d_2, d_1, d_0] = [1000, 200, 30, 4]$. Therefore, it can be stated that $1,234 = 1000 + 200 + 30 + 4$. The magnitude stack is used to generate all possible forms of a given number.

Generating forms of a number

At this stage, all the possible *Igbo* forms of a number were then derived by some special combinations of close elements of its magnitude stack. The results were pushed into the forms stack. For example, the magnitude stack for 1,255 is $[d_3, d_2, d_1, d_0] = [1000, 200, 50, 5]$ and the possible forms and generation for this number are shown in Tables 4 and 5

respectively. The number of possible forms largely depends on the values of d_3 , d_2 , d_1 and d_0 . The next step is the decomposition of the elements of the form stack to a form containing only the basic lexical items, the multiplicative base, and the forms of subtraction to give the surface representation of the forms. In these forms, '-' represents subtraction (*Bere*), '+' represents addition (*na*) within a number phrase, while '- -' and '+ +' represents subtraction (*Bere*) and addition (*na*) between number phrases respectively.

Table 2. Magnitude generator algorithm

```

1 initialize number, remainder as integer
2 integer d0, d1, d2, d3 ← 0
3 integer divisor ← 10
4 stack magnitude_stack ← empty stack
5 while number ≠ do
6 remainder integer number % divisor
7 if remainder ≠ 0
8 Magnitude stack push(remainder)
9 number ← number-remainder
10 while mag in magnitude stack do
11 if mag < 10
12 d0 ← d0+mag
13 else if mag <100
14 d1 ← d1+mag
15 else if mag <1000
16 d2 ← d2+mag
17 else mag >= 1000
18 d3 ← d3+mag
19 magnitude_stack ← [d0, d1, d2, d3]
20 return magnitude_stack
    
```

Table 3. Magnitude Stack of some Numbers

Number	d_3	d_2	d_1	d_0	Magnitude stack
13	0	0	10	3	[10, 3]
45	0	0	40	5	[40, 5]
125	0	100	20	5	[100, 20,5]
418	0	400	10	8	[400, 10, 8]
625	0	600	20	5	[600, 20,5]
1255	1000	200	50	5	[1000, 200, 50,5]

Table 4. Forms of *Igbo* grammar

Forms	Derivation
1.	[d_3, d_2, d_1+d_0]
2.	[d_3, d_2+d_1, d_0]
3.	[d_3, d_2, d_1, d_0-10]
4.	[$d_3, d_2+d_1+d_0$]

Table 5. Generation of forms of 1,625

Derivation	Form Stack	Decomposition Stack
d3,d2,d1+d0	[1000,600,25]	[1000+100*6+10*2++5]
d3,d2+d1,d0	[1000,620,5]	[1000+100*6++10*2+5]
d3,d2,d1,d0-10	[1000,600,20,-5]	[1000+100*6+10*2-5+10]
d3,d2+d1+d0	[1000,625]	[1000+100*6++10*2++5]

The Context Free Grammar (CFG) that was formulated for the *Igbo* numeral system is described as follows:

```

NUMBER ::= PHRASE (NUMBER|APHRASE) | APHRASE .....1
APHRASE ::= PHRASE.ADD .....2
PHRASE ::= DIGIT|M PHRASE .....3
DIGIT::={1,2,3,4,5,6,7,8,9}
M::={10,100,1000}
    
```

Rules 1 and 2 are interpreted by addition, rule 3 is interpreted as multiplication. NUMBER can be a PHRASE only or PHRASE and NUMBER. DIGIT describes the lexicon which serves as building blocks for the *Igbo* numeral. DIGIT contains {*Otu, Abuo, Ato, Ano, Ise, Isii, Asaa, Asato, Itolu, Iri, Nari, Puku*}. M is a set of multiplicative base which are 10 (*Iri*), 100 (*Nari*) and 1,000 (*Puku*). NUMBER is the actual number to be transcribed. A NUMBER may be formed from several numbers giving the grammar a recursive form.

The addition sign (+) in the parsing of the number translates to 'na' when the sign is between the tens and units and it translates to coma ',' when the sign is between thousand and hundred or hundred and tens.

System Implementation

Python 2.7 was used for the system development both Desktop (Figure 3) and Android Applications (Figure 4). An object oriented programming (OOP) approach was used during the implementation with three modules, which include: basis.py: that contains the base numbers and their translation. The numbers stored here form the base in which other numbers were generated; generator.py houses the class generator and interface.py contains the code for the graphical user interface.

The two classes for each of the modules include: the generator and the interface. The generator performs the main translation tasks and were implemented using generateMaginitude(self,num), Parse(self,stack) and Translate(self,parseStack) methods. The interface (gui) is the second class for the modules which was implemented using createwidget(self), convertfunction(self) and Clearfunction(self) methods.

System Testing

Questionnaires were administered to fifteen (15) *Igbo* speakers as a means of testing and evaluating the system using random numbers between 1-1000 and the analysis, description of respondents as well as information were provided by each respondent. Age (ranges from 15 to 32 years), sex, state of origin, educational level and knowledge of *Igbo* numeral system were used as the metrics. The analysis is shown in Tables 6 and 7.

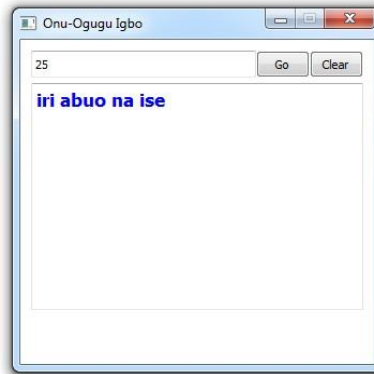


Figure 3. Result of number 25 for desktop application.

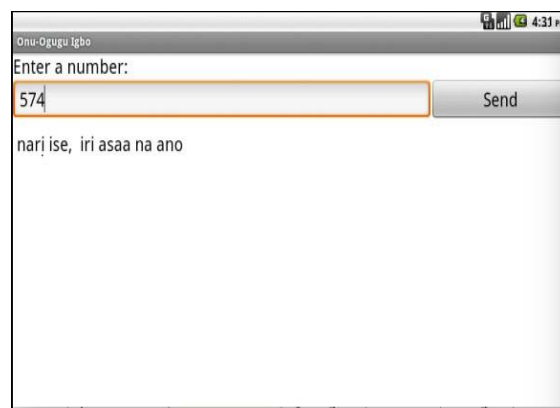


Figure 4. Result of Number 574 for Android Application

Table 6. Analysis of the description of the respondents

Knowledge	Score (%)	Frequency
Weak	20	2
Average	20	2
Adequate	30	3
Excellent	50	5

Table 7. Data of the knowledge versus score

Item	Scale	Respondents
sex	Male	9
	Female	6
Educational Level	SSCE	2
	Under Graduate	7
	Graduate	4
	Post Graduate	2
Knowledge of <i>Igbo</i> Numeral System	Weak	3
	Average	4
	Adequate	2
	Excellent	6

System Validation

An expert of *Igbo* language and a native of Imo State (South-Eastern part of Nigeria), who has an experience of teaching the language for more than ten (10) years and presently a teacher of the language at Obafemi Awolowo University International School (OAUIS), validated the system. Twenty (20) random numbers were used for the test and all the numbers were certified correct by the expert. The result of the validation from the expert is shown in Table 8. It can therefore be concluded that the system performs excellently and can be used as a learning tool for *Igbo* language numerals.

Conclusion

In this research, a study and analysis of the *Igbo* numerals with focus on extracting the knowledge needed and the arithmetic requirements for their representations has been carried out. The specific complexities found in the *Igbo* numeral system have also been discussed. The results of the study were presented and discussed with a view to providing a key to the most suitable *Igbo* representation for numbers.

The result shows that the *Igbo* number system has a systematic concept underlying it, which can be analysed using modern knowledge of mathematics and computing. In this study, it was found that the *Igbo* numeral system is decimal (base 10). The developed system gives 100% accuracy on the computation underlying the derivation of *Igbo* numerals. This system as has a place in effective teaching and learning of the *Igbo* language and can adapted to other languages.

Table 8. System Validation Results

S/N	Number	System's Translation	Comments
1.	35	<i>Iri ato na ise</i>	Correct
2.	125	<i>Otu nari, iri abuo na ise</i>	Correct
3.	202	<i>Nari abuo na abuo</i>	Correct
4.	465	<i>Nari ano, iri isi na ise</i>	Correct
5.	708	<i>Nari asaa na asato</i>	Correct
6.	814	<i>Nari asato, iri na ano</i>	Correct
7.	423	<i>Nari ano, iri abuo na ato</i>	Correct
8.	347	<i>Nari ato, iri ano na asaa</i>	Correct
9.	589	<i>Nari ise, iri asato na itolu</i>	Correct
10.	111	<i>Otu nari, iri na out</i>	Correct
11.	1	<i>Otu</i>	Correct
12.	10	<i>Iri</i>	Correct
13.	15	<i>Iri na ise</i>	Correct
14.	25	<i>Iri abuo na ise</i>	Correct
15.	40	<i>Iri ano</i>	Correct
16.	146	<i>Otu nari, iri ano na isii</i>	Correct
17.	999	<i>Nari itolu, iri itolu na itolu</i>	Correct
18.	854	<i>Nari asato, iri ise na ano</i>	Correct
19.	676	<i>Nari isii, iri asaa na isii</i>	Correct
20.	1000	<i>Puku</i>	Correct

Although this work has provided an assessment of the *Igbo* numeral system, however, more peculiarities may exist in the *Igbo* numeral system which have not been captured within the scope of this study.

References

- Ablanedo, J., Aiken, M. and Vanjani, M. (2007) Efficacy of english to spanish automatic translation, *Int. J. Information and Operations Management Education*, 2(2):194–210.
- Akinadé, O. O. (2014). 'Formal Analysis and Computational Implementation of a Number to Yoruba Text Translation System'. Unpublished M.Sc. Thesis, Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife.
- Craciunescu, O., Gerding-Salas, C. and STRINGERO'KEEFFE, S. (2004). Machine translation and computer-assisted translation. *Machine Translation and Computer-Assisted Translation*.
- Ekundayo, S. A. (1977). Vigesimal Numeral Derivational Morphology: *Yoruba* Grammatical Competence Epitomized. *Anthropological Linguistics*, 19 (9), 436-453.
- Everett, D. (2005). Cultural constraints on grammar and cognition in Pirahã. *Current anthropology*, 46(4), 621-646.
- Eves, H. W. (1969). *An introduction to the history of mathematics*. Holt, Rinehart and Winston, Newyork, 3rd edition.

- Hammarstron, H. (2009). Rarities in Numeral Systems. In Wohlgemuth, J. and Cysouw, M, Editors, *Rara& Rarissima: collecting and interpreting unusual characteristics of human Languages; Empirical Approaches to languages Typology*, 7-55. Mouton de Gruyter.
- Hutchins, J. (2005). Current commercial machine translation systems and computer-based translation tools: system types and their uses. *International Journal of Translation*, 17(1-2): 5-38.
- Knight, K. (1997). Automating knowledge acquisition for machine translation. *AI Magazine*, 18(4), 81.
- Knight, K., and Graehl, J. (1998). Machine transliteration. *Computational Linguistics*, 24(4), 599-612.
- Mbah, E. E., Ahamefula, N. O. and Ayegba, M. (2014). A Comparative Analysis of the Numerals and Counting Systems of Igbo and Igala. *Research on Humanities and Social Sciences*, 4 (26).
- Ogomaka, P.M.C, (2005). Traditional *Igbo* Numbering System: A Reconstruction. Published Thesis, Department of Evaluation and Research Methods, University of Nigeria, Nsukka.
- Sanusi, I. O. (1995). Counting in Base Five: The Derivation of Numerals in B`at`on`u. In *Nigerian Language Studies*, 3, 13 – 19.
- Wilder, R.L., 1973, *Transworld Student Library: Evolution of Mathematics Concepts*, London: Transworld Publishers.
- Yusuf, O. (Ed.). (2007). *Basic linguistics for Nigerian languages teachers*. Linguistic Association of Nigeria.
- Zaslavsky, C. (1973). *Africa Counts*. Prindle, Weber & Schmidt. Inc., Boston, Mass.