Finite State Representation of Reduplication Processes in Igbo

[1]Olamma Iheanetu, [2] Michael Adeyeye,
[1]Africa Regional Centre for Information Science (ARCIS), University of Ibadan, Nigeria
olaneee@yahoo.com

[2]Department of Information Systems, Cape Peninsula University of Technology, Cape Town
adeyeyem@cput.ac.za

***Abstract*—This paper presents a finite state model of reduplication processes in Igbo. Identified Igbo reduplication processes are based on (i) verbal reduplication with prefixation, (ii) total nominal reduplication (iii) ideophone reduplication. However, this work identifies and includes a fourth and fifth type that occurs in the language, namely; (iv) adverbial reduplication and (v) prepositional reduplication. Xerox Finite State Tool (XFST) was used in representing the five Igbo reduplication processes computationally. Igbo verbal reduplication exhibits selective reduplication process and is characterized by prefixation and vowel replacement. Vowel harmony phenomenon was taken into consideration in achieving verbal reduplication to cater for phonological changes. Model testing results showed 84% accuracy in both analysis and recognition of reduplicated forms in Igbo.**

***Index Terms*— Reduplication, Igbo Language, Finite State Automaton, XFST, Morphology**

1. INTRODUCTION

Morphology is the lowest level of linguistic analysis. It is concerned with the rules of word formation in a language. Derived from the Greek word *morphe*, morphology implies shape or form [18]. [18] further defined morphology as the arrangement of members of an object and how these members combine to form a whole. In linguistic sense, an object signifies a word, while its parts or members refer to its constituent morphemes. Therefore, it is logical to define morphology as a branch of linguistics concerned with the morphotactics of a language, its realization from morphemes and applications. The study of morphology can either take a traditional (by human linguists) approach or computational approach.

Computational morphology is used to describe a formal representation or study of natural languages using Natural Language Processing (NLP) techniques in order to achieve an analysis unique to that particular language. It can be classified into concatenative and non-concatenative morphology. In simple terms, computational morphology is concerned with word processing either in text or speech formats [19]. Several tools and techniques subsist for the study of computational morphology. However, computational morphology tasks are challenging for computers. The approach adopted in this study is Finite State Techniques (FST). This owes to the fact that any regular expression can be encoded into a finite state network or automata. This makes it possible to accommodate the infinite vocabulary inherent in natural languages.

Finite State Morphology (FSM) is an approach to morphological description (of natural language) using regular expressions [19]. It is based on finite state transitions from one state to another in representing the morphotactics of a language. This representation can also be via regular expressions. Finite state morphology is a very popular approach to rule-based and supervised learning of natural language morphology. This may be due to a shared similarity with natural language behaviour. The potentially infinite vocabulary of a language can be captured in a Finite State Automaton (FSA) using lexicons and continuation classes or via regular expressions. Therefore, it is then possible that a FSA, that represents a particular language based on the morphological rules of the language modeled into the automaton, is capable of generating vocabulary of that language which may not currently be in use in that particular language. This is the idea of generative grammar rule. The generative grammar rule stipulates that an infinite vocabulary of a language can be generated from a finite set of rules. Although FSTs have proven to be very efficient in the description of concatenative morphology, they are limited in handling non-concatenative morphology; a characteristic of Bantu languages, of which Igbo is one.

In this study, we have given a brief description of Igbo morphology and a discussion on the three identified reduplication processes based on [16]. In addition, we discuss a fourth and a fifth type of Igbo reduplication namely; adverbial and prepositional reduplication. A detailed description of the morphotactics of these reduplication processes is given, as well as a formal representation of reduplication in Igbo using finite state descriptions and regular expressions. We conclude in this study by identifying the limitations and challenges encountered. This study is an aspect of an on-going research.

2. ABOUT IGBO LANGUAGE

Igbo is a tonal language spoken by approximately twenty million native speakers in Nigeria [15], a country in the Western part of Africa, and has over thirty dialects. Igbo belongs to the Kwa family of Niger Congo, and exhibits high and low tones that apply different meaning to the same phone sets [7]. Igbo is highly agglutinative [20] and inflects for tense and mood. In addition, there exists a third tone in Igbo phonology, called a down-step. Only vowels and nasals are tone-marked in Igbo. High and low tones are marked with grave accents while a hyphen or bar is used to mark the down-step.

Identified concatenative morphological processes in Igbo are nominalization and verbal inflections, which makes the representation of Igbo morphology using LEXICONS and continuation classes an easy task in finite state notations. However, other morphological processes exhibited in the language like compounding, interfixing, circumfixing and reduplication pose a challenge in its representation using LEXICONS. This is because these processes are non-concatenative while LEXICONS are modeled to capture concatenative morphology. [2] agrees that non-concatenative morphology is a challenge, but also the cutting edge of computational morphology.

An important concept that impacts on Igbo morphology is the concept of vowel harmony. [19] defines it as a phonologically-based concept where the leftmost vowel in a word afftects the other vowels following it in that word. Specifically, vowel harmony is a feature stipulating that only vowels belonging to the same group can co-occur [6], [22], [10] in a word. Two vowel groups exist in Igbo language, namely;

(i) heavy vowel group {e, i, o, u}
(ii) light vowel group {a, ị, ọ, ụ}

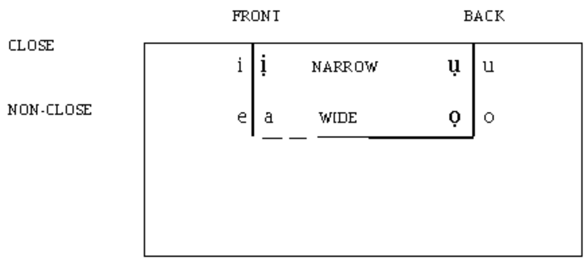

Figure 1 - Igbo vowel harmony group chart (Emenanjo, 1987)

2.1. REDUPLICATION PROCESSES IN IGBO

Reduplication is a non-concatenative process of word formation. [12] defines it as a non-concatenative phonological concept. In general, reduplication can be viewed as a morphological process that involves the repetition of a morpheme or an entire word, which sometimes results in phonological changes in the new word. When an entire word or morpheme is reduplicated, it is known as full stem reduplication. *Selective, partial* or *limited* reduplication are terms used to describe a type of reduplication that involves reduplicating only a part of the morpheme or word. [15] identified three types of reduplication in Ọ̀gbahụ̀ morphology and we extend it to standard Igbo (SI), namely - (i) verbal reduplication with prefixation, (ii) total nominal reduplication (iii) ideophone reduplication. This study includes (iv) adverbial and (v) prepositional reduplication processes in the list. Igbo nominal, adverbial, prepositional and ideophone reduplication processes undergo full stem reduplication. No form of phonological changes occurs. Igbo verbal reduplication (with prefixation) displays selective reduplication. Verbal reduplication in Igbo is a very productive process that yields gerunds, that is, verbal nouns. Some forms of phonological changes are observed in this process. [15] also stresses that recurring forms in Igbo serve both grammatical and semantic functions.

A. Nominal Reduplication in Igbo

For the purposes of this study, we break down nominal reduplication into three forms. We state that one way of achieving nominal reduplication in Igbo is by the duplication of the end result of the nominalization process. That is, nominalizing a CV verb by prefixation, and reduplicating the resulting noun. This definition is different from that given in our reference literature by [16]. Another form of nominal reduplication occurs with numerals but they do not undergo nominalization. A third category of nominal reduplication involves underived nominal words. These are nouns that cannot be broken down. Nominal reduplication is categorized as full stem reduplication. Mostly they carry semantic information and are sometimes used for emphasis. Example 1 below shows nominal reduplications that are realized by duplicating the end result of a nominalization process.

Example 1
a) ọ̀nwụ̀ (death) → {ọ̀nwụ̀}^2 → ọ̀nwụ̀ọ̀nwụ̀ (death death)
b) urè (rot) → {urè}^2 → urèurè (roten roten)
c) nri (food) → [nri}^2 → nrinri (food food)

Example 2 below typifies nominal reduplication from nouns that cannot be broken down further.

a) ndọ̀rọ̀ (drag/tussle) → {ndọ̀rọ̀}^2 → ndọ̀rọ̀ndọ̀rọ̀ (tussle).
b) ezè (teeth) → {ezè}^2 → ezèezè (teeth teeth).
c) nsọ (holy) → {nsọ}^2 → nsọnsọ (holy holy)
d) ike (power) → {ike}^2 → ikeike (power power)

Numeral reduplication connotes some semantic sense where used (Maduagwu, 2006). This is shown in example 3 below.
a) otu (one) → {otu}^2 → otuotu (one by one)
b) iri (ten) → {iri}^2 → iriiri (ten by ten/ in tens)
c) abụọ (two) → {abụọ}^2 → abụọabụọ (two by two/ in twos)

B. Ideophone Reduplication

Ideophones are words that contain no semantic information, but are representative of the way they sound. From this definition, it can be deduced that idiophone reduplication makes no semantic sense. Like the nominal reduplication, idiophone reduplication is characterized by full stem reduplication. Example 4 below elucidates what idiophone reduplication in Igbo looks like.

a) kọi → {kọi}^2 → kọịkọị (the sound of shoe heels on the floor during motion)

b) kpọm → {kpọm}^2 → kpọmkpọm (the sound of a knock on the door)

c) gbam → {gbam}^2 → gbamgbam (roofing sheets)

C. Verbal Reduplication with Prefixation

Verbal reduplication in Igbo (with prefixation) is a productive process that yields gerunds. Gerundive formations in Igbo are realized from the reduplication of verbs of the CV construct with prefixation. There are three ways of realizing gerunds in Igbo. However, only one approach is described in this study.

1) Verbal reduplication with close vowels. In this type of verbal reduplication, the stem is repeated and a low tone ò or ọ̀ is prefixed with respect to vowel group of the occurring stem vowel. Example 5 below typifies verbal reduplication (with prefixation) with close vowels. That is, the reduplication of a CV verb base results in a new word – ò / ọ̀CVCV.

Example 5

a) lụ → ọ̀ + {lụ}^2 → ọ̀lụlụ (marry)

b) □ụ → ọ̀ + {□ụ}^2 → ọ̀□ụ□ụ (drinking)

c) bi → ò + {bi}^2 → òbibi (home/ place of living)

d) zụ → ọ̀ + {zụ}^2 → ọ̀zụzụ (train)

e) ju → ò + {ju}^2 → òjuju (fullness / satisfaction)

f) yi → ò + {yi}^2 → òyiyi (resemblance)

g) pị → ọ̀ + {pị}^2 → ọ̀pịpị (squeeze/ press)

See http://en.wikipedia.org/wiki/Igbo_language for Igbo labials.

. *D. Adverbial Reduplication*

Adverbial reduplication is a full stem reduplication involving doubling of the adverb stem without any vowel deletions or assimilation. Example 6 highlights some adjectival reduplication including illustrations of their usage in Igbo sentences.

a) nwayọ (slow) → {nwayọ}^2 → nwayọọnwayọ (slow slow)

Obi na-aga nwayọnwayọ → Obi is walking slow slow → Obi is walking slowly

Obi na-eme nwayọnwayọ → Obi is doing slow slow → Obi is gentle

b) mpe (tiny) → {mpe}^2 → mpempe (tiny tiny)

O gburu anụ ahu mpempe → He cut the meat tiny tiny → He cut that meat in tiny bits

c) gburu (round)→ {gburu}^2 → gburugburu (round about/surround)

Ahịa gbara ụlọ ha gburugburu → Market run round about their house →The market surrounds their house

E. Prepositional Reduplication

Just like adverbial reduplication, prepositional reduplication also involves full stem reduplication. Often, they make a semantic sense where used, and this sense could get ambiguous some of the times in some scenarios.

Example 7 – prepositional reduplication

a) elu (up) → {elu}^2 → eluelu (up up).

Ọ na-ebu onwe ya eluelu → He carries himself up up → He is proud.

b) akụkụ (side)→ {akụkụ}^2 → akụkụakụkụ (side side)

Onye oshi ahu ji akụkụakụkụ banye n'ụlọ Obi → That thief used side side and entered Obi's house → That thief stealthily entered Obi's house.

c) ime (inside) → {ime}^2 → imeime (inside inside).

Ọ na-ekwu imeime → He is talking/ speaking inside inside→ He is talking/ speaking faintly.

2.2 COMPUTATIONAL APPROACHES TO REDUPLICATION

From a review of literature, computational studies in Igbo are still at a fledgling stage, although modest efforts have been made by some researchers. [17] and [1] have carried out computational studies in Igbo on machine readable dictionary for Igbo text and Igbo morphological analyzer, respectively. However, none of these studies have attempted reduplication process in Igbo. [1] concentrated on verbal realizations and nominalizations, highlighting linguistic data; a scare resource for less developed and understudied languages, as a major limitation in his study. He utilized finite state machines in the analysis and generation of Igbo verbal forms that share inflectional properties. The study accommodated a total of 141 verb stems. Further literature search did not yield any other computational studies in Igbo

2.3 FINITE STATE APPROACH TO REDUPLICATION

Amidst the challenges of finite state approaches to handling reduplication, many studies [3], [5] still adopt it because of its proven success in handling the concept of reduplication. [19] indicated that a major demerit of finite state lexicon is redundancy and inability to cope with words not identified in the lexicon. Formal representation of reduplication phenomenon in **xfst** poses a challege to developers because **xfst** is concatenative in nature. However, this challenge has been addressed by the **compile-replace** algorithm embedded in **xfst.** This algorithm, according to [2] has successfully handled full stem reduplication and interdigitation of Semitic languages better than [13] **two-level formalism**. However, the algorithm comes with an increase in the memory demands of the resulting network.

[9] supports the use of **compile-replace** algorithm to handling reduplication in Bantu languages. The study proposes this approach over the finite state concatenation which yields copies of same verbs in different forms in the resulting network. [21] argues that the compile replace-

algorithm by [2] is not a mathematically based solution. He therefore proposes the description of reduplication of natural languages which involves enriching the finite state automata (FSA) with operations like *repeat*, *skip*, and *self loops*. These new operations will allow an easy representation of difficult language phenomenon like non-concatenative process in finite state networks (FSN). For instance, the *repeat* arcs will allow backward movement within a string and thus repeat a part of it (to model reduplication). *Skip* arcs allow forward movements in a string while suppressing the spell out of some of its letters while *self loop* arcs can be utilized in the modeling of infixation. However, [9] focus was on Prosodic morphology.

Owing to the fact that FSNs lack memory to store necessary information while traversing the network, [4] propose a finite-state registered automata (FSRA), that can augment FSA with memory in the form of registers. This eliminates the need for duplicated paths as the automaton becomes equipped with memory to remeber symbols. Each arc of the network is associated with registers for either a reading or writing action. The read action, denoted as *R*, allows traversing an arc only if a designated register contains a specific symbol while the write action *W*, allows traversing an arc while writing a specific symbol into a designated register.

Another proposal for handling different types of reduplication manifest in various languages, using FST was made by [8]. Because of the complexity introduced by curent methods in handling reduplication, the study suggests simpler notations of handling contraints imposed by reduplicating strings of a language using operators that can bring about equivalence of discontinuous strings.

3. Methodology

Finite state transitions can not sufficiently handle reduplication because the continuation classes which the LEXICONS utilized in **lexc** were not built to accommodate non-concatenative morphological processes like reduplication. To tackle this challenge, **compile-replace** algorithm; a regular expression statement was incorporated in the **xfst** toolkit. This algorithm caters for non-concatenative processes like reduplication [2]. **Compile-replace** algorithm has been proven to yield better results than [13] **twolc** rules for the same purpose. [2] further stated that **compile-replace** algorithm is also very efficient in handling interdigitization (a non-concatenative process) in Semitic languages like Arabic. On this premise, this study adopts the **compile-replace** algorithm for our model description. This approach, according to [9], is efficient for handling reduplication in Bantu languages, of which Igbo is one. In addition, [3] and [14] applied finite state techniques to successfully represent reduplication processes in Gikuyu and Vietnamese respectively.

3.1 IMPLEMENTATION

A distinct network was built to describe four types of reduplication processes identified in Igbo using REs in **xfst**. Three of these processes involving verbal reduplication can be described as selective reduplication while the last type is full stem reduplication. Out of the five identified grammatical classes that accept reduplication in Igbo, four of these classes undergo full stem reduplication while only one class, the verbal class, undergoes selective reduplication. In handling the full stem reduplication, the solution presented by [2] was used to implement adverbial, ideophone, nominal and prepositional reduplications. This involves the concatenation of the original word and the duplicated part (that is, a copy of the original stem) to form a new word. This new word is written without a hyphen or white space in-between, in Igbo.

In this study, **xfst** script was utilized in the description of full stem reduplication in Igbo. In the script IgboRedup; Igbo consonants were grouped into two classes and defined as LC (labial consonants) and NLC (non-labial consonants). This grouping was necessitated by the peculiarity of the process involved in the formation of verbal nouns (gerunds) in Igbo. Conversely, the vowels had to be represented by two classes HV (heavy vowels) and LV (light vowels) signifying the two distinct vowel groups in the language. With these definitions in place, it was then easy to enforce restrictions on Igbo morphotactics in order to realize valid gerunds in Igbo.

IgboRedup, contains two distinct networks that represents the full stem reduplications and verbal reduplication with prefixation. These networks were composed into a single network (regular expression). One of the grammars 'Igbo' describes four grammatical classes that accept reduplication in Igbo. Hence we have the noun, adverbial ideophone and prepositions defined by 'Igbo'. Due to the fact that gerundive formations which involve verbs take on a slightly different approach to reduplication, they were defined by a different grammar, 'Igbo1'. Verbs accept prefixation and an optional reduplicating vowel which is based on the class of the consonant of the verb stem in gerundive realizations.

The reduplication domain was marked by the delimiters “^[“ and “^]” in both grammars. It was necessary to define Igbo reduplication procedure in IgboRedupHV and IgboRedupLV respectively for heavy and light vowels in the grammar because these groups accept different prefixes for verbal reduplication with respect to vowel harmony. Likewise Igbo1RedupHV and Igbo1RedupLV define the same procedure but this time, only catering for the verbs. As earlier stated these two grammars were composed into a single regular expression and unioned in order to provide the option of either grammar. After the composition of the grammars, the invocation of the **compile-replace** algorithm identifies and copies the reduplication segment marked out by the delimiters “^[“ and “^]”. It is the part of the network surrounded by the delimiters that is reduplicated and removes the delimiters.

4 TESTING AND RESULTS

The excerpt below shows an analysis and generation result when the regular expression grammar IgboRedup was tested for analysis and generation.

```
xfst[1]: xfst[1]: apply up
apply up> ureure
ure+Grnd
apply up> òriri
ri+Grnd
```

```
apply up> òyiyi
yi+Grnd
apply up> mpempe
mpe+Grnd
apply up>
xfst[1]: apply down
apply down> mpe+Grnd
mpempe
apply down> mpe+Verb
mpe
apply down> yi+Grnd
òyiyi
apply down> yi+Verb
yi
apply down> ri+Grnd
òriri
apply down> ri+Verb
ri
```

In order to account for the accuracy of the system, the system was tested with a new word set for analysis and generation. Out of this, 84% of the words were correctly reduplicated while the remaining 16% were wrongly reduplicated. Close observation showed that most of the words that were wrongly reduplicated had the consonant ‘ch’. The reason for the lack of recognition of this consonant is not yet known, although other diagraph consonants in the language are recognized by the system.

5. LIMITATIONS AND CONCLUSION

This is part of an on-going study to represent Igbo morphology computationally. In this study, we have achieved full stem and selected reduplication, as it occurs in Igbo language, using XFST. This includes nominal, ideophonic, adverbial, prepositional reduplication and verbal reduplication (with prefixation). Future work will involve an extension of the study to include verbal reduplication (with prefixation and vowel replacement) and other identified non-concatenative morphological processes in the language like interfixation, circumfixation and compounding. In addition, tones and morphophonological changes will be included. This will make the model truly representative of Igbo morphology. For clarity, examples of verbal reduplication (with prefixation and vowel replacement) that we hope to consider in future are:

a) ga → ọ̀gịga (go)
b) ke → òkike (creation)
c) chọ → ọ̀chịchọ(search)
d) me → òmume (action)
e) fọ̀ → ọ̀fụfọ̀ (squeeze)
f) fu → òfufu (loss)
g) gwụ → ọ̀gwụgwụ (end/ exhaustion)

We acknowledge lack of adequate linguistic resources as a major limitation to this study. More so, the **xfst** version (version 8.1.4) used for this project does not support utf-8 characters and this was a major challenge in this study. Igbo language distinguishes between the two vowel groups which may never co-occur in a single word. As a way around this limitation, we used the upper case versions of these alphabets in place of the utf-8 characters. That is, 'ị' was replaced with 'I', 'ọ' with 'O' and 'ụ' with 'U'. Also the Igbo nasal '□' was replaced with 'N' for same reasons. An additional limitation is that words containing the consonant 'ch' was not recognized by **xfst** although other diagraphs like 'gb', 'gh', ‘gw’ 'kp', 'kw' were recognized.

REFERENCES

[1] Ayogu, I., Adetunmbi, A., Alese, B., and Kammelu, N. (2011). Igbomorph: A finite State-Based Morphological Analyser for Igbo. In: AICTTRA 2011 Proceeding, pp. 1-12.

[2] Beesley, K. R and Karttunen, L. (2003). Finite State Morphology. Stanford, United States of America. CSLI Publications.

[3] Chege, K.; Ng’ang’a, W.; Wagacha, P. W.; De Pauw. G.; and Mutiga, J. (2011). Morphological Analysis of Gĩkũyũ using a Finite State Machine. In: Proceedings of Conference on Human Language Technology for Development, Alexandria, Egypt, 2-5 May 2011.

[4] Cohen-Sygal, Y. and Wintner, S. (2006). Finite-State Registered Automata for Non-Concatenative Morphology. Computational Linguistics Vol 32 (1).

[5] Dalrymple, M., Liakata, M., and Lisa Mackie (2005). A Twolevel Morphology of Malagasy In: Proceedings of PACLIC 19, the 19th Asia-Pacific Conference on Language, Information and Computation

[6] Emenanjo, N (1987). Elements of Modern Igbo Grammar. Ibadan. University Press Limited (UPL).

[7] Gale Group Inc. (1999). Igbo. In: *Junior Worldmark Encyclopedia of World Cultures* (2009). Encyclopedia.com. (Internet). Retrieved from http://www.encyclopedia.com/doc/1G2-3435900354.html. Accessed on 20th August 2008.

[8] Hulden, M. and Bischof, S. T. (2010). Annotating Reduplication in Finite-State Morphology.

[9] Hurskainen, A. (2005). Reduplication in Bantu Languages. In Arppe et al. (Eds.), Inquiries into Words, Constraints and Contexts.

[10] Iloene, G. (2007). Igbo Phonology. In O. Yussuf (Eds.), Basic Linguistics for Nigerian Languages Teachers (pp. 163 - 180). Port Harcourt, M & J Grand Orbit Communications Ltd and Emhai Press.

[11] Kaplan, R., D. (2001). The Science of Plant Morphology: Definition, History and Role in Modern Biology. American Journal of Botany, 88: 1711 – 1741.

[12] Katamba, F. (1993). Morphology. London. Macmillan Press Limited.

[13] Koskienniemi, K. (1983). *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production* (Doctoral thesis), Department of General Linguistics, University of Helsinki, Finland.

[14] Le, H. P., Nguyen, T. M. and Azim, R. (2009). Finite State Description of Vietnamese Reduplication. In: Proceedings of the 7th Workshop on Asian Language Resources held in Suntec, Singapore, 6-7 August, 2009, pp.63–69.

[15] Linux, N. (2010). Igbo Open Source Translation Project. Retrieved from http://igbo.sourceforge.net/. Accessed on 19th October, 2010.

[16] Maduagwu, G. O. (2006). *Lexicalization Strategies in Ọ̀gbahụ Dialect in Igbo, Nigeria.* (Doctoral thesis), Department of Linguistics and African Studies, University of Ibadan, Nigeria, 135p.

[17] Ng'ang'a, W. (2010). Towards a Comprehensive, Machine-readable Dialectal Dictionary of Igbo. In: Proceedings of the 2nd workshop on Africa Languages Technology (AfLAT) held in Valleta, Malta, 18th May, 2010, pp.63-67.

[18] Ritchey, T. (2011). General Morphological Analysis. Retrieved from http://www.swemorph.com/ma.html. Accessed on 22nd November, 2011.

[19] Trost, H. (2003). Computational Morphology. Retrieved from http://www.coli.uni-saarland.de/~schulte/Teaching/ESSLLI-06/Referenzen/Morphology/trost-2003.pdf. Accessed on 24th august 2012.

[20] University of California, Los Angeles. (2009). Igbo. In: *Language Materials Project*. Retrieved from http://www.lmp.ucla.edu/Profile.aspx?LangID. Accessed on 20th October, 2009.

[21] Walther, M. (2000). Finite-State Reduplication in One-Level Prosodic Morphology. In: proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics. Retrieved from http://www.aclweb.org

[22] Zsiga, E. C. (1992). A Mistmatch Between Morphological and Prosodic Domains: Evident from Two Igbo Rules. Phonology, 9(1), 101-135. doi: 10.1017/S0952675700001512