Development of Grapheme-to-Phoneme Conversion System for *Yorùbá* Text-to-Speech Synthesis

ABÍMBỌ́LÁ R. ÌYÀNDÁ [1]
ỌDẸ́TÚNJÍ A. ỌDẸ́JỌBÍ[1]
FESTUS A. SOYỌYE[2]
OLÚGBÉNGA O. AKINADÉ

Ọbáfẹ́mi Awólọ́wọ̀ University, Ilé-Ifẹ̀, Ọ́sun State, Nigeria[1,2].
University of the West of England, Frenchay Campus, Bristol BS16 1QY [3]
Computer Science & Engineering Department[1]
Foreign Languages Department[2]
Bristol Enterprise Research and Innovation Centre (BERIC) [3]
[1] (abiyanda,oodejobi)@oauife.edu.ng
[2] fsoyoye@oauife.edu.ng
[3] agbenga28@gmail.com

Abstract. Grapheme-to-Phoneme (G2P) conversion is concerned with describing the process of automatic conversion of letters in a text into their phonemic transcription and it plays an important role in text to speech synthesis. The G2P component of *Yorùbá* TTS is yet to be addressed. To achieve the conversion of the grapheme to phoneme, knowledge of the process underlying standard *Yorùbá* text to sound is expedient. This paper presents a system for generating the phonemic description of the sound corresponding to a piece of *Yorùbá* text. Standard *Yorùbá* text data was collected and digitised and resulting digital text was edited for correction of orthographic items using Tákàdá text editor and Àkọtọ́ *Yorùbá* software. The corresponding speech data for the text collected was recorded in a quiet environment with a noise cancelling microphone on a typical multimedia computer system using the Speech Filing System software and analysed and annotated using PRAAT speech processing software. The system was designed with Finite State Transducer and implemented using Python 2.7 programming language. The system was able to generate multiple correct representations for graphemes in *Yorùbá* language with 100% accuracy on both grapheme level and word level. The results obtained in this study confirmed the hypothesis that the *Yorùbá* G2P system has a systematic procedure underlying it and this procedure can be computationally specified, analysed and represented using computational tools.

Keywords: Grapheme-to-Phoneme, Text-to-Speech, Standard *Yorùbá*., Tone Languages

(Received August 18th, 2014 / Accepted February 27th, 2015)

1 Introduction

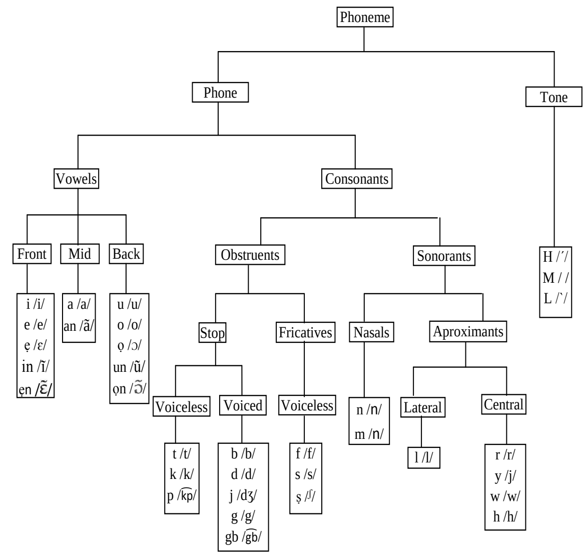
Grapheme-to-Phoneme conversion (G2P) is a crucial component of Text-to-Speech (TTS) that is concerned with finding pronunciations of words and assigning prosodic structure (such as phrasing, intonation and duration) to them [6].

The term Grapheme-to-Phoneme (G2P) conversion is commonly used for describing the process of automatic conversion of letters in a text into their phonemic transcription and plays an important role in text to speech synthesis. It determines the correct sequence of phonemes for a particular word when presented with that word orthography [24]. It is also essential for limiting the size of the lookup lexicon and for relying on an automatic conversion in preference to a large lexicon if it is worthwhile to trade off memory footprint versus correct pronunciation [20].

A system for converting graphemes to their phonemic transcription is an important tool in Natural Language Processing especially in high level speech processing as well as Speech Recognition and Spoken Dialog Systems development in which its primary goal is to accurately predict the pronunciation of a novel input word given only the spelling. Attempts have been made to develop other components of Standard *Yorùbá* TTS ([21], [22], [23]), but the G2P subsystem is yet to be addressed. To achieve the conversion of the grapheme to phoneme, knowledge of the process underlying standard *Yorùbá* text (SY) to sound is expedient. This study aimed at developing a system for generating the phonemic description of the sound corresponding to a piece of *Yorùbá* text and to establish that *Yorùbá* G2P system has a systematic concept underlying it which can be specified, analysed and represented using computational tools.

Grapheme is the smallest semantically distinguishing unit in a written language while a phoneme is an entity that is a transcription of a sound which is nearer to the pronunciation of a word than its orthographical representation. Phoneme that occurs in tone languages uses tone to convey differences in lexical meaning. For example, in Standard Yorùbá (SY) language, bisyllabic words *ọkọ́ (hoe), ọ̀kọ̀ (spear), ọkọ (husband), ọkọ̀* (vehicle) are phonemically identical but correspond to different lexical meaning. In summary, the SY phoneme can be classified according to sound as shown in Figure 1.

The rest of this paper is organized as follows: Section 2 gives an overview of *Yorùbá* Language, Section 3 briefly discusses G2P conversion approaches. Section 4 summarises the related works, Section 5 describes the design and implementation of *Yorùbá* G2P conversion system and Section 6 discusses the result. Section 7 gives the system evaluation. The work is summarized in the final section.



Note: Mid tone when on syllabic nasal is marked with dash (-)

Figure 1: Classification of SY Phoneme according to Sound

2 Standard *Yorùbá* Language

Yoruba language is one of the three major indigenous languages, along with Hausa and Igbo in Nigeria and it is spoken by over 37 million people [10]. Yoruba is a native language of the Yoruba people, an ethnic group primarily located in south-western Nigeria (Lagos, Oyo, Ogun, Ondo, Ekiti, Osun and parts of Kwara and Kogi states). Standard *Yorùbá* (SY) language is used in language education, the mass media and everyday communication. The SY alphabet has 25 letters which is made up of 18 consonants which represented graphemically by *b, d, f, g, gb, h, j, k, l, m, n, p, r, s, ṣ,t, w, y* and seven vowels represented graphemically by *a, e, ẹ, i, o, ọ, u* [1]. It should be noted that the *gb* is a diagraph i.e. a consonant written with two letters. There are five nasal vowels in the language represented graphemically as *an, ẹn, in, ọn, un* and two syllabic nasals represented graphemically as *n* and *m*. It should also be noted that *an* and *ọn* are phonemically the same. The consonant and vowel systems as well as a more detailed description of SY are presented in [16].

2.1 Standard *Yorùbá* Syllable

Syllables are the smallest unit of pronunciation in a language [2]. syllables are considered as the basic unit of sound because it has been established as a perceptually and acoustically coherent unit [1]. In SY language, there are five syllable structures. This is made up of oral vowels (V), nasal vowels (Vn), syllabic nasals (N), combination of consonants and oral vowels (CV) as well as combination of consonants and nasal vowels (CVn) [21].

The syllables are the tone bearing elements of the SY language. For example, in the statement: *Abím̄bọ́lá ti lọ sí oko* (*Abím̄bọ́lá* has gone to the farm), there are ten syllables. This also implies that there exist ten tones (MHMHH-M-M-H-MM).

2.2 G2P approaches

The issue of mapping textual content into phonemic content is highly language dependent. There are three main approaches of G2P conversion and these are: data driven approach ([11], [13], [15], [5], [14]), rule based approach ([25], [29], [8], [27]) and hybrid ([28].

The data driven approach uses a data driven model whereby the mapping from grapheme to phoneme is constructed automatically by processing a training set. This approach falls into the general category of machine learning techniques. It was pointed out that the operation of many data driven approaches is no different from the phonological rules; the questions stored in a decision tree could be reformulated as context-sensitive rules [20]. In data-driven approach, the algorithm is learned automatically from data.

According to [9], rule-based systems are a good choice when one has only a few data samples, when one has experts that are readily available for consultation, or when one finds that an explanation for answer is essential. Rule-base is modular in nature, in that it allows for modification. New rules can be added and old ones can be deleted, since rules are independent of one another.

Rule based approach are important computational tools and have been used for a variety of purposes including word or name lookups for database searches and speech synthesis. Such rules are also necessary to formalize grapheme-phoneme correspondences in speech synthesis architecture.

While the use of a dictionary is more important now that denser and faster memory is available to smaller systems, letter-to-sound still plays a crucial and central role in speech synthesis technology. However, for some languages with complex writing systems such as Chinese and Japanese, building the rules is laborious and very difficult to cover most possible situations [12]. This approach was adopted in this study, *Yorùbá* being a resource-scarce language.

Hybrid approach combines both rule-based and data-driven approach together in solving the TTS problem.

3 Related work

[27] presented a Sinhala Grapheme-to-Phoneme Conversion and Rules for Schwa Epenthesis. The paper established that finding correct pronunciation for a given word is one of the first and most significant task in linguistic analysis process. Kiran et al. (2001) presented a model-based approach to improving grapheme to phoneme conversion as well as phoneme to grapheme conversion to train oral reading skills in two patients with severe oral reading and naming deficits.

The result shows that grapheme to phoneme conversion treatment is useful in improving oral reading and oral naming skills.The experiment provides evidence for incorporating cognitive neuropsychological models in aiding the development of appropriate treatment protocols, and demonstrates the importance of rule-based learning, rather than compensatory strategies, in maximising the effects of generalisation.

[26] presented a G2P rule-based system Korean Grapheme to Phoneme Conversion based on sound patterns to convert Korean text strings into phonemic representations. A set of 20 sentences were used for the evaluation and the accuracy of the system is 98.70% on conversion. This shows that the rule-based sound patterns are effective on Korean grapheme-to-phoneme conversion problem.

[18] investigated the use of acoustic data to improve grapheme-to-phoneme conversion for name recognition by introducing a joint model of acoustics and graphonemes where graphoneme parameters can be estimated using a maximum. Results showed that the best performance came from a simple data combination strategy, which yielded a relative word error rate (WER)/sentence error rate(SER) reductionof 8.6%/7.2%. Applying discriminative training to ML-adapted model, enlarges the WER/SER reduction to 11.5%/11.9%.

[17] presented a technique for aligning the phonemes in a phonemic transcription of words with the graphemes in its orthographic representation. The alignment algorithm is driven by a table of received pronunciation phonemes and their corresponding orthography. The algorithm has been tested against 33,121 words of a phonemically tagged version of the Lancaster-Oslo/Bergen Corpus. Thirty-two thousand seven hundred and seventy-four words were aligned without error while three hundred and forty-seven words could not be aligned.

[3] presented a Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System and described a preprocessing morphological analyzer integrated into an Amharic Text to Speech (AmhTTS) System, to convert Amharic Unicode text into phonemic specification of pronunciation. This work focused on disambiguating gemination and vowel epenthesis which are the significant problems in developing Amharic TTS system. The evaluation test on 666 words shows that the analyzer assigns geminates correctly (100%).

Also, [7] presented a general framework for building letter to sound (LTS) rules from a word list in a language. The paper established that in order to make the building of models easier, a standardized alignment between the letters in an entry and the phones in its pronunciation is essential. The rules were applied to English (UK and US), French and German and some models were generated. The generated models achieve, 75%, 58%, 93% and 89%, respectively, from the word lists.

4 Data collection

Data collected for this study was from various sources such as textbooks and newspapers. Other texts from Internet sources were also used. Word units are defined as characters between white spaces. The texts were edited using a text editor (Tákàdá) and were corrected for graphemic items (tone marks and under dots) using Àkọtọ́ *Yorùbá*. Àkọtọ́ *Yorùbá* is a software developed by [4] for restoration of *Yorùbá* diacritics.

In this system, databases of pre-selected units (syllables) was used. The creation of these databases involved selecting a single unit, for inclusion in the system representing all possible occurrences. To cater for all possible segmental (auto and supra) contexts, the unit needs to be as neutral as possible.

The sample data collected is shown in Tables 2. The data is based on the *Yorùbá* syllable structure. The are about 687 syllables (V=7x3, Vn=5x3, N=1x3, CV=126x3 and CVn=90x3) collected. The concatenation of those syllables form a word and the concatenation of words produce a sentence. It should be noted that not all the syllables presented in Table 2 are valid *Yorùbá* syllables. For example, "*l*" cannot precede a nasal vowel (e.g. *lan, lẹn, lọn, lun,* does not exist in the *Yorùbá* words).

5 Design and Implementation of *Yorùbá* G2P conversion

The processes involved in the G2P for SY include: (i) Input tokenisation; (ii) Nasal characters process (e.g. to convert 'a' and 'n' to 'an' where appropriate); (iii) Syllabification process; and (iv) Graphemes to Phonemes conversion. To facilitate the process of converting grapheme to phoneme, the character of more than a single code-point (e.g. *ẹ̀, ẹ́, ọ̀, ọ́*), were transformed [4] as well as punctuations for internal processing as shown in Table 1.

Table 1: Code replacement for *Yorùbá* characters with more than one single code-points and punctuations

Serial-No	Character	Replacement
1	‘ ’	‘-’
2	‘ ’	‘- -’
3	‘-’	‘- - -’
4	‘,’	‘-,’
5	‘.’	‘-.’
6	‘?’	‘-?’
7	‘!’	‘-!’
8	u‘ẹ̀’	‘z’
9	u‘ẹ́’	‘x’
10	u‘ọ̀’	‘c’
11	u‘ọ́’	‘v’
12	‘gb’	‘q’

i. Input tokenisation: The input to this process is a text file. Tokenisation takes the input text and breaks it into sentences marked by a new-line, and each sentence is further breakdown into chunks called tokens. Tokenisation 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 is separated by a front slash (/).

ii. Processing Nasal Characters: This stage processes the nasal characters. The nasal characters are: *an, ẹn, in, ọn, un*. For example: to convert *'a'* and *'n'* to *'an'*; *'ẹ'* and *'n'* to *'ẹn'*; *'i'* and *'n'* to *'in'*; *'ọ'* and *'n'* to *'ọn'*; and *'u'* and *'n'* to *'un'* where appropriate.

iii. Syllabification Process: This process breaks words into syllables. The correct pronunciation of a word requires not only the phonemic transcription, but also the syllable boundaries [19]. Determining the syllable boundary in tone languages is very important because the strength of the tone on a vowel within a syllable is dependent on its position.

   At this stage, the syllabification algorithm breaks a word to the basic unit of speech. The algorithm dynamically looks for polysyllable units making up the word, cross checks the database for availability of units, and then breaks the word based on the syllable structures.

Table 2: *Yorùbá* syllable data for high tone

Consonant	**OralVowel (V)**							**NasalVowel (Vn)**					**SyllabicNasal (N)**
(C)	á	é	ẹ́	í	ó	ọ́	ú	án	ẹ́n	ín	ọ́n	ún	ń
C	**CV**							**CVn**					
b	bá	bé	bẹ́	bí	bó	bọ́	bú	bán	bẹ́n	bín	bọ́n	bún	
d	dá	dé	dẹ́	dí	dó	dọ́	dú	dán	dẹ́n	dín	dọ́n	dún	
f	fá	fé	fẹ́	fí	fó	fọ́	fú	fán	fẹ́n	fín	fọ́n	fún	
g	gá	gé	gẹ́	gí	gó	gọ́	gú	gán	gẹ́n	gín	gọ́n	gún	
gb	gbá	gbé	gbẹ́	gbí	gbó	gbọ́	gbú	gbán	gbẹ́n	gbín	gbọ́n	gbún	
h	há	hé	hẹ́	hí	hó	họ́	hú	hán	hẹ́n	hín	họ́n	hún	
j	já	jé	jẹ́	jí	jó	jọ́	jú	ján	jẹ́n	jín	jọ́n	jún	
k	ká	ké	kẹ́	kí	kó	kọ́	kú	kán	kẹ́n	kín	kọ́n	kún	
l	lá	lé	lẹ́	lí	ló	lọ́	lú	lán	lẹ́n	lín	lọ́n	lún	
m	má	mé	mẹ́	mí	mó	mọ́	mú	mán	mẹ́n	mín	mọ́n	mún	
n	ná	né	nẹ́	ní	nó	nọ́	nú	nán	nẹ́n	nín	nọ́n	nún	
p	pá	pé	pẹ́	pí	pó	pọ́	pú	pán	pẹ́n	pín	pọ́n	pún	
r	rá	ré	rẹ́	rí	ró	rọ́	rú	rán	rẹ́n	rín	rọ́n	rún	
s	sá	sé	sẹ́	sí	só	sọ́	sú	sán	sẹ́n	sín	sọ́n	sún	
ṣ	ṣá	ṣé	ṣẹ́	ṣí	ṣó	ṣọ́	ṣú	ṣán	ṣẹ́n	ṣín	ṣọ́n	ṣún	
t	tá	té	tẹ́	tí	tó	tọ́	tú	tán	tẹ́n	tín	tọ́n	tún	
w	wá	wé	wẹ́	wí	wó	wọ́	wú	wán	wẹ́n	wín	wọ́n	wún	
y	yá	yé	yẹ́	yí	yó	yọ́	yú	yán	yẹ́n	yín	yọ́n	yún	

The syllabification was achieved using specialised push down automata (PDA) with two stacks. The stacks are 'syllables' (syl) and 'buffer' (buf). 'syl' holds the syllable tokens while 'buf' is a temporary storage location. The system takes the test file and buffer it. The system based on the item that is on top of the 'buffer' stack, push the input item either to 'syllables' or 'buffer' or pop item 'buffer'. The syllabification function that capture the behaviour of this system is shown below:

```
function getSyllables(words){
  buffer = [];
  syllables = [];
  for(i, ch in enumerate(word)){
    if((ch in V) or (ch in Vn)){
      if(len(buffer) == 0){
        syllables.push(ch);
      }
      else{
        top = buffer.pop();
        if(top in C or top in N)
          syllables.push(top+ch);
        }
      }
    }
    else if(ch in C or ch in N){
      if(len(buffer) == 0){
        buffer.push(ch);
      }
      else{
        top = buffer.pop();
        if(top in N){
          syllables.push(top);
          buffer.push(ch);
        }
      }
    }
    else{
      syllables.push(ch);
    }
  }

  return syllables;
}
```

The PDA for the syllabification process is shown in Figure 2.

iv. Graphemes to Phonemes conversion: This stage converts all the graphemes resulting from the previous stage into phonemes. Different methods for building automatic alignments have been proposed, among which there are one-to-one and many-to-many alignments [5].

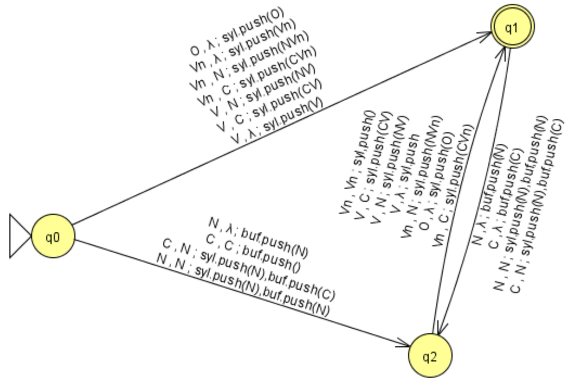


Figure 2: PDA for Yorùbá text syllabificator

Table 3: Code transform to original character

Serial-No	Character	Replacement
1	'z'	u'ẹ̀'
2	'x'	u'ẹ́'
3	'c'	u'ọ̀'
4	'v'	u'ọ́'
5	'q'	'gb'

In one-to-one alignment each letter corresponds only to one phoneme and vice versa. To match grapheme and phoneme strings, an "empty" symbol is introduced into either string. In many-to-many alignments a letter can correspond to more than one phoneme and a phoneme can correspond to more than one letter.

In this study, one-to-one alignment was adopted. This is achieved by first replacing all transformed characters to their original characters. For example, grapheme ẹ which was formally replaced with 'z' to facilitate the process of converting grapheme to phoneme was converted back to ẹ before the phonemic transcription output was given. Syllables with high, mid and low tones are transcribed with their tones (H, M, L, respectively) placed in bracket beside them.

6 Result Discussion

Figure 3 illustrates the main user interface. The interface contains two operations- (i) tokenize and (ii) syllabify and G2P. Tokenise option gives user opportunity to tokenise input statements, copy output to clipboard and save output to file. It also provides analysis of the token (stating the number of times a token exist) in the

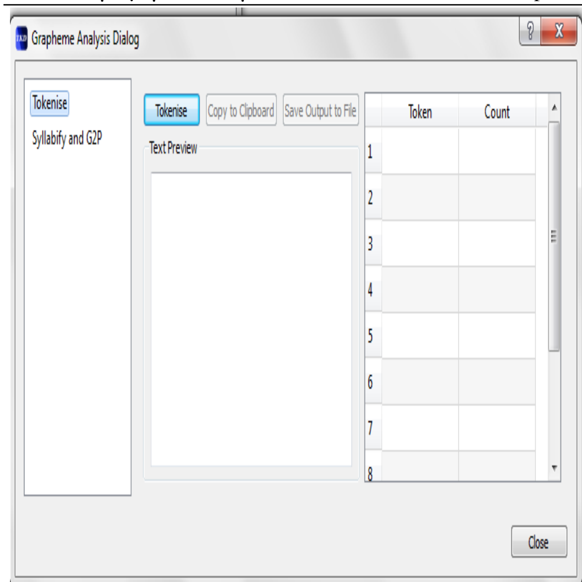


Figure 3: Interface for Yorùbá Grapheme-to-Phoneme system

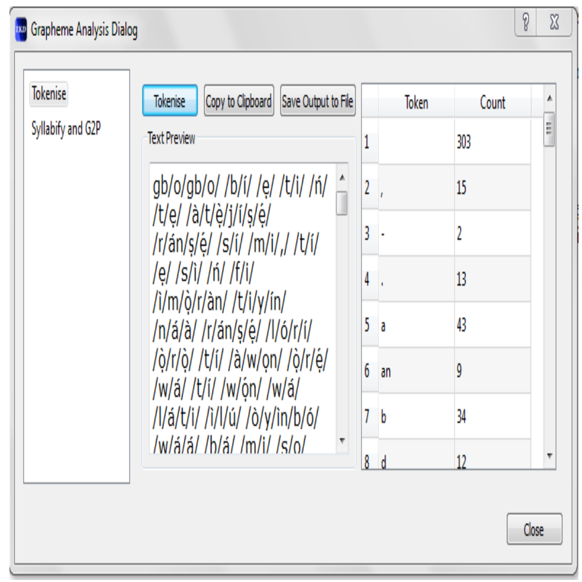


Figure 4: Screen shot for sentence tokenised

input statements. Syllabify and G2P option gives user opportunity to syllabify the input statements, convert all grapheme in the statements into their phonemic transcription and save output to file.

Figure 4 shows the output for Software processing of tokenisation. The total number of tokens for the selected statements are 1,175 as distributed in Table 4.

Figure 5 shows the output for Software processing of syllabification. Hyphen (-) is used as the syllable marker. The total number of syllables for the selected statements are 185 contributing to approximately 42%, 43% and 15% of it having high, mid and low tones, respectively. The distribution is shown in Table 5.

Figures 6 shows the output for Software processing of G2P conversion. All graphemes were correctly transcribed to the phonemic description. This is in line with the fact that a grapheme in *Yorùbá* corresponds to exactly one phoneme. The associated tone of each phoneme is placed in bracket beside each phoneme.

7 Evaluation

The aim of the evaluation is to compare the performance of the developed system with that of experts. To accomplish this, the accuracy of the system to predict the data was investigated. This was done to ascertain the correctness of the transcription done by the developed system using

Generally, according to [25], the main transcription errors can be divided into three: quantitative errors; qualitative errors and character errors

Table 4: *Yorùbá* token count

S/N	Token	Count	S/N	Token	Count
1		303	31	à	44
2	,	15	32	àn	5
3	-	2	33	á	47
4	.	13	34	án	4
5	a	43	35	è	7
6	an	9	36	é	19
7	b	34	37	ì	23
8	d	12	38	ìn	2
9	e	4	39	í	47
10	f	11	40	ín	4
11	g	4	41	ò	11
12	gb	9	42	ó	17
13	h	2	43	ù	5
14	i	43	44	ú	8
15	in	2	45	ún	7
16	j	20	46	ń	9
17	k	22	47	ǹ	3
18	l	32	48	‘	2
19	m	24	49	ḿ	1
20	n	30	50	ṣ	13
21	o	23	51	ẹ	10
22	p	10	52	ẹn	1
23	r	34	53	ẹ̀	12
24	s	22	54	ọ	19
25	t	37	55	ọn	9
26	u	3	56	ọ̀	24
27	un	5	57	ọ̀n	1
28	w	38	58	ọ́	9
29	y	22	59	ọ́n	4
30	ùn	1	60	ẹ̀	17

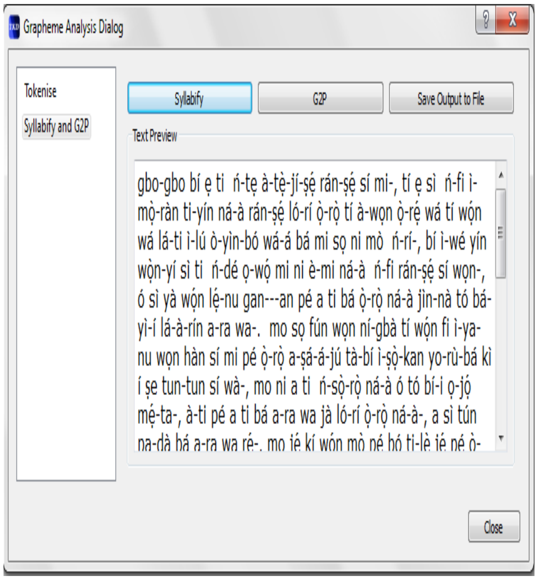


Figure 5: Screen shot for sentence syllabified

Table 5: Syllable analysis

S/N	Tone	Frequency	Proportion
1	High	199	38.42
2	Mid	181	34.94
3	Low	138	26.64
	Total	518	**100.00**

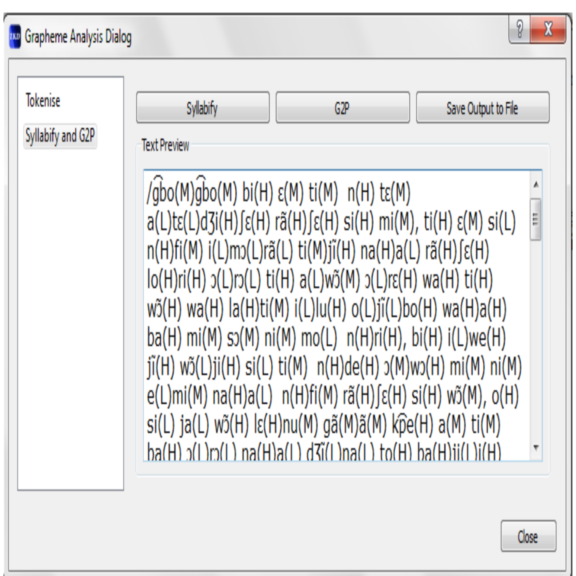


Figure 6: Screen shot for sentence transcribed

Table 6: Evaluation result for *Yorùbá* G2P system

	Test Text
No. of words	302
No. of correctly transcribed word	302
No. of erroneous transcription	0
Correct rate word level	100%
No. of grapheme	1213
No. of correctly transcribed grapheme	1213
No. of erroneous transcription	0
Correct rate phoneme level	100%

i. Quantitative errors: where a vowel is transcribed as long instead of short, or vice versa.

ii. Qualitative errors: where a vowel is transcribed as rounded instead of unrounded or vice versa.

iii. Character error: where a wrong character is used in the transcription.

Character error was encountered in G2P conversion from the system. This error occurred when there was a wrong input for the IPA equivalents of two (*j* and *y*) of the graphemes during the system implementation. For every word that contains *j* and *y*, the system gives the transcription for *j* and *y* as ɟ and *y*, respectively instead of ʤ and *j*.

For example, formally, the system gave the transcription for *Yorùbá* as *jo(M)ru(L)ba(H)* instead of *jo(M)ru(L)ba(H)* and for *jèrè* as *je(L)re(L)* instead of ʤe(L)re(L). This issue was addressed by replacing the *j* and *y* with ʤ and *j* respectively for the phonemic representation.

Also, the system encountered error in syllabifying words that has two syllabic nasal *n* following each other. Examples is found in *nǹkan*. The two *n's* are taken as a syllable, thereby syllabifying the word *nǹkan* as *nǹ-kan* (having two syllables instead of three). Two occurrences which were recorded were addressed. An algorithm for the nasal vowel identification was written for its Syllabification Process.

Two male human expert transcriptions were used to evaluate the developed system output through the use of a questionnaire. The respondents were asked to provide the phonemic transcription of some *Yorùbá* statements from the language resource created for use in this research. The transcription of this system gave 100% accuracy as shown in Table 6. This implies that the level of intelligibility of the system is 100%. However, the tones (H, M, L) are represented by putting it in bracket beside each word transcribed instead of being marked on the phoneme.

8 Conclusion

In this work, the study and analysis of the *Yorùbá* texts with focus on extracting the knowledge needed developing a G2P system based on *Yorùbá* text have been carried out. The specific challenges encountered in implementing the *Yorùbá* G2P system have also been pointed out.

The results of the study have been presented and discussed with a view to provide a key to the most suitable *Yorùbá* transcription for texts. The result shows that the *Yorùbá* G2P system has a systematic concept underlying it and this concept can be specified, analysed and represented using computational tools. Although this work provided a holistic assessment of the *Yorùbá* G2P system, it is also assumed that more peculiarities may exist in the *Yorùbá* G2P system that has not been captured in this study.

References

[1] Adewole, L. O. *The Categorical Status and Function of the Yoruba Auxiliary Verb with some Structural Analysis in GPSG*. PhD thesis, University of Edinburgh, Edinburgh, 1988.

[2] Àkànbí, L. A. and Odéjobí, O. A. Automatic Recognition of Oral Vowels in Tone Language: Experiments with Fuzzy Logic and Neural Network Models. *Applied Soft Computing*, 11(1):1467–1480, 2011.

[3] Anberbir, T. and Gasser, M. Grapheme-to-phoneme conversion for amharic text-to-speech system. *Conference on Human Language Technology for Development (HLDT), Alexandria, Egypt*, pages 68–73, 2011.

[4] Asahiah, F. O. *The development of a Standard Yorùbá Diacritics Restoration System*. PhD thesis, Obafemi Awolwowo University, Ile-Ife, Nigeria, 2014.

[5] Bisani, M. and Ney, H. Joint-Sequence Models for Grapheme-to-Phoneme Conversion. *Speech Communication*, 50(5):434–451, 2008.

[6] Black, A. W. and Lenzo, K. Building voices in the festival speech synthesis system: Processes and issues in building speech synthesis voices. available from http://www.festvox.org. pages 77–80, 2000.

[7] Black, A. W., Lenzo, K., and Pagel, V. Issues in Building General Letter to Sound Rules. *3rd ECWA Workshop on Speech Synthesis*, pages 77–80, 1998.

[8] Black, A. W. and Lenzo, K. A. Building synthetic voices. available from festvox.org/bsv/bsv.pdf. 2007.

[9] Caudill, M. Expert networks. *Byte*, 16(10):108–110, 1991.

[10] CIA. Cia world factbook 2014. Available online at http://www.cia.gov/cia/publications/factbook., 2014. Visited April, 2014.

[11] Damper, R., Marchand, Y., Adamson, M., and Gustafson, K. A comparison of letter-to-sound conversion techniques for english text-to-speech synthesis. *Proceedings of the Institute of Acoustics*, 20(6):1998, 1998.

[12] Divay, M. and Vintale, A. J. Algorithms for grapheme-phoneme translation for english and french: Applications for database searches and speech synthesis. *Computational Linguistics*, 23(4):495–523, 1997.

[13] Fukada, T., Yoshimura, T., and Sagisaka, Y. Automatic generation of multiple pronunciations based on neural networks. *Speech Communication*, (27):63–73, 1999.

[14] Galescu, L. and Allen, J. F. Bi-directional conversion between graphemes and phonemes using a joint n-gram model. *4th ISCA Tutorial and Research Workshop ITRW on Speech Synthesis*, 2001.

[15] Huang, J., Olorenshaw, L., Hernández-Ábrego, G., and Duan, L. Memory efficient grapheme-to-phoneme conversion system for speech processing. *in Proc. ICSLP, (Jeju, Korea)*, 1:1237â–1240, 2004.

[16] Iyanda, A. R. *Design and Implementation of a Grapheme-to-Phoneme Conversion System for Yorùbá Text-to-Speech Synthesis*. PhD thesis, Obafemi Awolowo University, Ile-Ife, Nigeria, 2014.

[17] Lawrence, S. G. C. and Kaye, G. Alignment of phonemes with their corresponding orthography. *Computer Speech and Language*, 1:153–165, 1986.

[18] Li, X., Guanawardana, A., and Acero, A. Adapting grapheme-to-phoneme conversion for name recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 130–135. Institute of Electrical and Electronics Engineers, Inc., December 2007.

[19] Loots, L. Data-driven augmentation of pronounciation dictionaries. master of science in engineering thesis. Technical report, Stellenbosch University, Available from http://www.ams.sunysb.edu, 2010.

[20] Moore, P. J. Statistical grapheme to phoneme conversion using language origin. master of philosophy. Technical report, Hughes Hall, University of Cambridge, Available from http://www.ams.sunysb.edu, 2006.

[21] Odéjobí, O. A. A quantitative model of yorùbá speech intonation using stem-ml. *INFOCOMP Journal of Computer Science*, 6(3):47–55, 2007.

[22] Odéjobí, O. A. Design of a text markup system for yorùbá text to speech synthesis applications. *Conference on Human Language Technology for Development, Alexandria, Egypt*, pages 74–80, 2011.

[23] Odéjobí, O. A., Wong, S. H. S., and Beaumont, A. J. A modular holistic approach to prosody modelling for standard yorùbá speech synthesis. *Computer Speech and Language*, 22(1):39–â68, 2008.

[24] Taylor, P. Hidden markov models for grapheme to phoneme conversion. *Engineering*, 9(5):2–5, 2005.

[25] Torstensson, N. Grapheme-to-phoneme conversion, a knowledge-based approach. *TMH-QPSR Vol. 44 - Fonetik 2002*, pages 117–120, 2002.

[26] Wang, Y.-C. and Tsai, R. T.-H. Rule-based korean grapheme to phoneme conversion using sound patterns. In Kwong, O., editor, *PACLIC*, pages 843–850. City University of Hong Kong Press, 2009.

[27] Wasala, A., Weerasinghe, R., and Gamage, K. Sinhala grapheme-to-phoneme conversion and rules for schwa epenthesis. *Working Papers*, pages 500–507, 2007.

[28] Wester, M. Pronunciation Modelling for ASR - Knowledge-based and Data-derived Methods. *Computer Speech and Language*, 17:69–85, 2003.

[29] Yoon, K. and Brew, C. A linguistically motivated approach to grapheme-to-phoneme conversion for korean. *CSL*, 20(4):357–381, 2006.