Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 3 No. 2, June. 2015

The Digital Innovations & Contemporary Research in Science & Engineering Journal is an International Journal of
The Multidisciplinary Research Group
International Centre for IT & Development, USA

Article Citation Format
Imeh Umoren, Moses Ekpenyong & Ubong Etuk (2015): Grapheme-To-Phoneme, Phoneme-To-Grapheme (G2P-P2G) Transcription Machine for Tone Languages: A Database Approach.. Journal of Digital Innovations & Contempt Res. In Sc., & Eng Vol. 3, No. 2. Pp 1-12

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 191rd March, 2015
Review Type: Blind
Word Count Post Review: 4029
Review/Acceptance Information Sent : 12th May, 2015
Final Acceptance:: 29th May, 2015
DOI Prefix: 10.22624

Grapheme-To-Phoneme, Phoneme-To-Grapheme (G2P- P2G) Transcription Machine for Tone Languages: A Database Approach.

[1]Imeh Umoren, [2]Moses Ekpenyong & [3]Ubong Etuk
[1&3]Department of Computer Science, Akwa Ibom State University, Uyo, Nigeria
[2]Department of Computer Science, University of Uyo, Uyo, Nigeria
[1]hollymehumoren@gmail.com, [2]mosesekpenyong@uniuyo.edu.ng, [3]etuk_ub@yahoo.com
[1]+2348036813637

ABSTRACT

Grapheme-to-Phoneme, Phoneme-to-Grapheme (G-2-P and P-2-G) Transcription Machine for tone language is a challenging but often a necessary task. The implementation of tonal ascents is an artificial Intelligence (AI) complete. This research paper describes in details, modules required for the development of a good quality grapheme-to-phoneme (G-2-P) transcription for text-to-speech synthesis in Ibibio. We consider a (G-2-P) transcription machine for tone language which accept as input, the text of a language in (English format) and transcribe the affected sounds (fonts) to orthography/IPA format (equivalent). The specification of a general template is made in a character separated value (CSV format) for the different sound and present result for Ibibio, a Nigerian (West African) language. The template/Database is adaptable and can be expanded to accommodate more sounds and fonts. The machine will not only provide relieve to language secretaries/writers who spend hours or days struggling with conventional keyboards and language symbols but will also serve as a useful input for resolving language specific issues requiring close collaboration between linguists and Software Engineers.

Keyword: G-2-P, Artificial Intelligence (AI) Complete, Orthography/IPA and Ibibio

1. INTRODUCTION

We describe various modules which allow for the need to build a grapheme-to phoneme conversion system for Ibibio. Modules include a syllabification program, a fast morphological parser, a lexical database, a phonological knowledge base, transliteration rules, and phonological rules. Knowledge and procedures were implemented accordingly. Speech synthesis systems (SSS) consist of a linguistic and an acoustic part. The linguistic part converts an orthographic representation of a text into a phonetic representation flexible and detailed enough to serve as input to the acoustic part. The acoustic part is a speech synthesizer which may be based on the production of allophones or diphones. This paper is concerned with linguistic part of speech synthesis for Ibibio (a process we will call *phonomisation).*

The problem of phonemisation has been approached in different ways. Recently, connectionist approaches [21] mid memory-based reasoning approaches [22] have been proposed as alternatives) the traditional symbol-manipulation approach. Within the latter (rule-based) approach, several systems have been built for English (the most comprehensive of which is probably [1], and systems for other European Languages are beginning to appear. Text-to-speech systems for *Ibibio* are still in an experimental stage, and two different designs can be distinguished. Some researchers adopt an 'expert system' pattern matching approach, Boot, 1984, others a 'rule compiler' approach, [15]; [3] in which the rules are mostly in an SPE inspired format. Both approaches take the grapheme/phoneme as a central unit. We will argue that within the symbol manipulation approach, a modular architecture with the syllable as a central unit is preferred.

Indeed, Grapheme-to-phoneme (G2P) conversion is an important component to the emergence of Ibibio Text-to-speech (ITTS) system. In most of the alphabetical languages such as English, the main challenges of G2P module is to generate correct pronunciations for words that are out of vocabulary (OOV). However, unlike the OOV problem, the difficulty in Ibibio G2P conversion is to pick out one correct pronunciation from several candidates according to the context information such as part of speech, lexical words or position of the polyphone in a word or sentence.

Traditionally, the commonly used method is to list polyphonic words and characters with correct pronunciations into a dictionary. But such dictionary can not completely solve G2P problem, pronunciation rules according to the context are needed to handle more complicated problem.
Recently, various data-driven methods including Neural Network, (NW) Decision Tress (DD), Pronunciation-by-Analogy Models (PBAM), and Extended Stochastic Complexity Methods (SCM) have been tried to solve G2P problem. In this research paper, several algorithms are proposed to solve G2P problem in Ibibio language Transcription Machine (ILTM). As automatic rule base machine, it is hope to be efficient and widely used in numerous tasks, including the tonal transcription model on focus. This paper is leverage to solve the polyphone problem and also will receive great improvements.

Grapheme-to-Phoneme (G2P) conversion has been faced from the very beginning of speech technology research and any existing speech system implements its own language – dependent solution. What is bringing renewed interest on the research is the recent move towards multilingual systems. Systems that should be able to deal with several languages should preferably rely on language-independent engine acting on distinct linguistic knowledge bases. Nevertheless, the time needed to develop a new language should be reduced as far as possible overcoming the difficulty of acquiring a deep insight into language-specific phonetic features.

Indeed, automatic language – independent approach to G2P conversion are highly desirable. However, some work in this direction has already been reported in other languages. This research work was motivated by the need to providing an efficient environment to implement G2P converter for Ibibio language in Nigeria. Various resources are proven useful for the proposed G2P and P2G transcription System. These resources include: (i) the design of electronic dictionaries and (ii) the creation of local language speech synthesizers.

The goal of this paper is to implement a much more comfortable, affordable and less time consuming approach/alternative to handling/processing language symbols and diacritics insertion, which has always been a notorious nightmare to language secretaries/writers. The task is accomplished by building

(i) a P2G parser/engine,
(ii) a tone marking parser/engine and
(iii) exceptions (words in context modification, homophones/homographs, tonal contrasts, loan words and words not in dictionary) handler.

To achieve the above stated goal, the first step is to ease the typing of materials. Here we recommend that the text(s) of such materials be typed using the Speech Assessment and Phonetic Alphabet (SAMPA) format, which can easily be assimilated by the typists/secretaries and are available on the common keyboard(s), i.e. a special keyboard design is not necessary. The SAMPA notations could be modified to suit the ergonomic needs of a language as done for ìbìbìò in [11]. Secondly, the input text is passed to the LAPTOL engine and processed word after word. If a word exists in the existing dictionary/lexicon, the word is replaced with its tone-marked equivalent (i.e. itON → itọn). Otherwise the word is parsed using the p2g rules defined by the parser to convert the phoneme(s) to their grapheme(s) equivalent (for instance itON → itọn). But if the p2g rule matches no IPA symbol within the word, then the word is ignored (not attended to). The extent of correctness of the output text will depend on the level of exhaustiveness of such a dictionary /lexicon. Lastly, we handle the exceptions (tonal contrasts, words not in dictionary, homophones/homographs, etc.).

The advantages of this approach include:

i. Reduction in processing cost
ii. Elimination of service time delay: i.e. processing orthographic texts using the SAMPA notations and then performing the grapheme translation and tone-marking automatically, making the translation details transparent to the users.
iii. The provision of a meta-structure and adaptability of present features for other/similar tone languages.
iv. Enabling the assessment of the level of exhaustiveness of existing electronic dictionaries.

Adaptability of the LAPTOL will be achieved by defining a generic meta-structure of the phoneme-to-grapheme (p2g) translation template/database and dictionary/lexicon.

2. THE GRAPHEME TO PHONEME TASK

In general, the G2P task amounts to converting a grapheme stream into a phonetic one. One major by-product of this task is that when embedded in a TTS system, it is a step in the conversion of a text into a suitable representation for the speech synthesizer. Text is preprocessed in order to gain knowledge of its syntactic structure, expand numbers and acronyms, delimit words and locate their lexical stress. For some languages, in addition, morphemes or components of compound words must be identified.

Each grapheme word (or morpheme) is then transcribed into a corresponding phoneme string. After that, further rules are applied accounting for allophonic changes at word boundaries.

In Automatic Speech Recognition (ASR) systems, G2P is generally confined to the task of providing a reference transcription for the words in the recognizer vocabulary. Both in its TTS and ASR applications, the core of G2P conversion is the transcription of a single grapheme word. The traditional options for word G2P conversion as mentioned earlier are explicit rules or lexicon look-up or a combination of the two. Rules would rewrite single graphemes or grapheme dusters into the corresponding phonemes, depending on their context.

A phonetician or a computational Linguist is required to design the rules, through a time-consuming and knowledge intensive effort. While for highly regular languages this task is feasible, for others, such as English, huge lexica are necessary to account for exceptions. When a high coverage phonetically transcribed lexicon of inflected forms is available, the alternative is to obtain phonetic transcription by lexicon look-up. In this case, no deep knowledge of the language is required, but ad-hoc implementation in storing and accessing the lexicon is needed to meet real-time constraint. The drawback of this choice is that the system has no generalization capability; any word outside the lexicon cannot be transcribed.

A natural way to achieve the generalization constraints while compacting lexical information is to extract transcription rules from the lexicon in an automatic manner using some machine learning techniques.

3. G-2-P CONVERTER MODEL

The system has been developed to consist of a rule-processing engine, which is language independent. Language specific information is fed into the system in the form of lexicon, rules and mapping. The architecture of the G2P is shown in figure 2. This G-2-P framework has then been customized for Ibibio. The default character to phone mapping is defined in the mapping file. The format of the mapping is shown below and explained sub-sequently.

3.1 Character Type Class Phoneme

Character: The orthographic representation of the character.
Type: Type of the character, e.g. C (for Consonant), V (for Vowel), etc.
Class: The Class to which the character belongs. These class labels can be effectively used to write a rule representing a broad set of character.
Phoneme: The default phonetic representation of the character.

Specific contexts are matched using rules. The system triggers the rule that best fits the current context. The rule format is shown below:

$$\alpha_1 \alpha_2 \ldots \alpha_m \{\beta_1 \beta_2 \ldots \beta_m$$

Where α_i Class label of the i^{th} character as defined in the character phone mapping. Together, these α_s represent the context that is being matched.

β_j jth action specification node. Each such node has the form:

Action_Type: Pos: Phoneme_Str

Action_Type This field specifies the type of this action node. Possible values are K (keep), R (replace), I (insert) and A (append).

Pos The index of the character which is covered by the context of the rule (1 < Pos < m).
Phoneme_Str: Phoneme string used by this action node.

3.2 Designs Concepts

The G2P first looks for each input word, in the exception lexicon. For the Ibibio G2P, this lexicon is the phonetic compound word lexicon which is generated using the algorithm as indicated. If the word is present in the lexicon, the phonetic transcription of the input word is taken from the lexicon itself. Otherwise, the G2P applies the rules on this word and produces the phonetic transcription.

3.2.1 Algorithm implementing G2P Conversion

Step1: Construct a Grammar Class
Class grammar
{
define grammar rules
.
.
.
}

Step 2: (define LAPTOL (input-text)(
(for each text-input-word (curr-word)(
Cond ((if there exist a unique candidate in lexicon
(replace curr-word with equivalent phonemic Character from lexicon))
((if no unique candidate exist and word contains phoneme(s) that require p2g
(call p2g (curr-word)- replace phoneme(s) in current word with IPA equivalents))
(else (call excep-h(curr-word) – exception handler)))))))

The exception handler scheme is illustrated below:

(define excep-h (curr-word) (
(if word has tonal contrasts
(replace word with available phoneme transcription variants from lexicon)
(else (add word to exception lexicon)))))

Step 1 ⟶ Step 2: *Parse input text (Grapheme form) to conversion class*
Class G2P
{
1. *Open input file*
2. *call grammar class and test each word against the grammar rules*
3. *Mark wrongly formed words*
4. *Convert well formed words to orthographic (Phoneme) form (using the SAMPA/Orthographic Consonant/Vowels table or database)*
5. *Go to 2*

}

4. ILTM DESIGN

The design of the proposed language transcription machine will complement the keyboard. We will first discuss the technical issues: both hardware and software and present a simpler, workable, reliable and dependable implementation of the LAPTOL for ìbìbìò.

4.1 Technical Issues

4.1.1 Hardware:

The keyboard is the most common hardware peripheral that can assist in the input of data into the computer for processing. To solve the problem of language symbols/diacritics, there have been recent attempts to design special language keyboards for tone languages. [21] for instance have designed a South African keyboard with a layout and extensions of various fonts for Venda. The keyboard covers all the eleven official languages of South Africa. A recent groundbreaking achievement is the development of regional multilingual keyboards for some languages in the United States, South America, Nigeria and European Nation. The Kọnyin keyboard [22] extends the QWERTY keyboard by creating dual-shift keys (four shift keys) that enables users to add accents to letters directly.

This innovative approach (designing physical keyboards) has the following limitations/disadvantages:

(i) Mix up of shortcuts/key combinations, which are difficult to memorize, awful at first sight. This will still contribute to time delay for large corpora.
(ii) More alphabets on the keyboard.
(iii) How will new language symbols be accommodated? Because keyboards are static and customized for a language or few languages, the addition of new symbol(s) will require further key combinations or addition of more physical keys to the layout. Imagine if [22] were to accommodate five countries languages, we would have had five shift keys, etc.
(iv) Could be difficult to port and adapt customized ones for other languages,
(v) What about design and configuration complexities? The third point also adds o these complexities.
(vi) Design cost: may be expensive at the long run. Current cost of the Kọnyin keyboard may be quite expensive.

4.1.2 Software

[23] defines a specification for a new keyboard layout to replace the "*de facto*" layout that is presently used in Finland and Sweden. The implementation is a software "keyboard driver", which presents the user with a character map where the user selects and work with the desired language fonts. This to a great extent requires huge manual effort during processing.

Hence, we analyse the software issues under the following points:

(i) *Fonts*: currently there exist general fonts such as *SIL fonts*: Charis SIL, Gentium SIL, SIL Doulos IPA, etc. which have different patterns of rendering- Which font format do we adopt? The Unicode font UTF-8 will be used to implement the proposed LAPTOL. The reason is the present the translations in a universally acceptable format that will be easier to retrieve and process without loss/misrepresentation of characters/renderings as common with HTML/XML formats/representation. We recommend that the fot symbols and diacritiecs in the resultant text be presenved in a Portable Document Format (PDF) before communication.
(ii) *Operating System(OS) Version*: The software for the design must be portable: i.e. could sill run perfectly on different OS: Windows, Linux, Macintosh, etc. the proposed LAPTOL will be developed in Perl, a rapid prototype language that is portable on most OS.

(iii) *How do we handle language symbols and diacritics?* The most common approach is manual: Insert the symbols and diacritics using the Insert symbol during text processing (for instance in MS Word), which is time consuming and very boring for large corpora/annotations. With the proposed implementation, translations will be done automatically.

The hardware option is rather a faithful re-duplication of existing language character maps, which are readily available in many software and Oss and will not be discussed further.

4.2 LAPTOL Design Components

The LAPTOL architecture design is as shown in Figure 1 below:

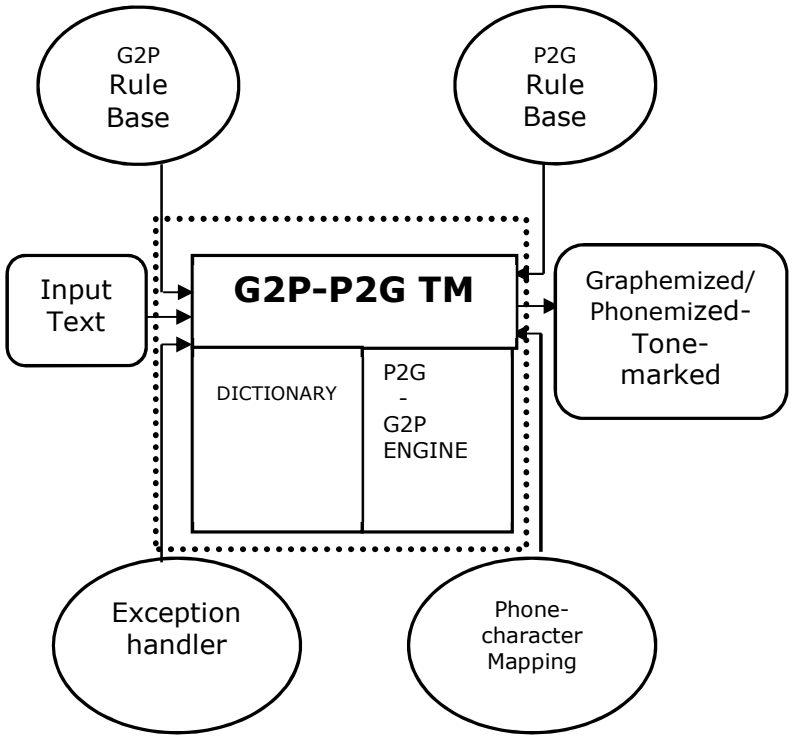

Fig 1. *Architectural model for the proposed LTM*

The design has the following components:

Input Text Interface: the input text could be typed in any word processing application, but should be saved as a text file (in text format), to enable Perl (our programming language choice) process the file. As earlier mentioned, the text should be typed using the SAMPA notation. Table 1 shows ìbìbìò graphemes and their SAMPA equivalent. The table has been modified to suit the ergonomic needs of ìbìbìò. The input text interface will get the prepared text in SAMPA and pass it to the language processor for processing.

Table 1: ìbìbìò Graphemes and their SAMPA equivalent. Source: [24].

S/NO	Graphemes	Sound Type	SAMPA
1.	A	Vowel	A
2.	Aa	Vowel	Aa
3.	B	Consonant	B
4.	D	Consonant	D
5.	E	Vowel	E
6.	E	Vowel	Ee
*7.	Ǝ	Vowel	X for @
8.	F	Consonant	F
9.	Gh	Consonant	R
10.	H	Consonant	H
11.	I	Vowel	I
12.	Ii	Vowel	Ii
*13.	ị	Vowel	I
14.	K	Consonant	K
15.	Kk	Consonant	Kk
16.	Kp	Consonant	Kp
17.	M	Consonant	M
18.	Mm	Consonant	Mm
19.	N	consonant	N
20.	Nn	Consonant	Nn
21.	Ny	Consonant	J
*22.	n̄	Consonant	N
*23.	n̄ ŋ̄	Consonant	NN
*24.	ʌ	Vowel	V
25.	O	Vowel	O
26.	Oo	Vowel	O
*27.	ọ	Vowel	O
*28.	ọọ	Vowel	OO
29.	P	Consonant	P
30.	Pp	Consonant	Pp
31.	S	Consonant	S
32.	T	Consonant	T
33.	Tt	Consonant	Tt
34.	U	Vowel	U
35.	Uu	Vowel	Uu
*36	ụ	Vowel	U for }
37.	W	Consonant	W
38.	Y	Consonant	J

To avoid conflict with Perl programming constructs and array symbols, we have replaced "@"- schwa with X and "}"- barred u with U. The only SAMPA equivalents the users need to know are marked with asterisks (*). These are notations for the IPA symbols; other letters are similar to their graphemic forms and are not characters combination.

5. LAPTOL IMPLEMENTATION AND EVALUATION

In this section, we discuss the implementation and evaluation of the designed LAPTOL. For this paper we ran the LAPTOL on "*The Tiger and the Mouse* "[1] in [17]. The implementation gave a fair enough assessment of the LAPTOL and we are refining the LAPTOL further to make it more intelligible. The components of the Laptol include: (i) *Perl scripts:* for mapping processes (p2g and tone marking) and exceptions handling, (ii) *A template/database:* for phoneme-grapheme translation entries and (iii) *an electronic dictionary:* a database of the language in comma separated Value (CSV) format using Open Office 2.0. we also include a user interface built in Visual Basic 6.0 for Windows OS users. The user interface request for an input file and informs the users where the output file is build.

The information section sets the path to the LAPTOL program by requesting for the directory that contains the LAPTOL. The user can then input the filename of the input text to be processed and click the *"process input text"* command button to process the text. The ***test*** input file for this paper was translated into ìbìbìò and is shown in figure 3.

ekpe ye usIne ekesaNa ke IkOt ekekIt eka isua uyo daNa ana ke isON. usIne ate, ekpe mbOk yak ami ndat ujo ami. NkO afo ukappa usuma uyo. fOn ido yem NkpO mfen dia. Ado ekpe amaasio mbara OmO OfVk ke uyo ate, ami nnie. muusOppOsOp idem udakka ke mmi, nya uta fiin ndian. usIne amaadakka ye ekamba mfVhO. ekpe amaanwana edimen afId uyo ado akaNkeed ado uyo ado amaakefaha enye ke usVNitON ndien domo daNa enye ekpedodomo ikikanna ifakka. akekemkoo, ewa amaadiboiyo, ekpe abeeNe unwam. "inieghe se Nkan nnam", ewa ObOOrO asaNa oboiyo. akema mfuOt aditamma oboiyo, ekpe ebeeNe unwam. "inieghe se Nkan nnam", mfuOt ObOOrO atamma oboiyo. ke akpatre, ekpe amaamiaNNa adikisIm nte usIne adVN. usIne mbon amaaketIppe isON OdVk asIne. "mbOk nnyaNa miin", ekpe ate. "uyo ado mmOdo amfaha ke unVNitON, ndien Nkanna mfakka". usIne ate, "afo ado ata idiOk unam. idXho ukuyakka nO nta uyo ado, ado nyaasVk unwam. NwaNNa inua mfo nO ntamma ndVk Nkwek uyo ado nO asIp ado se aboiyo udVk usVNitON". Ekpe amaanwaNNa inua, usIne OtOONO edikwuek uyo ado. Ekpe ekere ete, ObiON aanekke andON.

Fig.3. ìbìbìò translation of "The Tiger and the Mouse"-

The processed output is shown in Figure 4 on the nest page

[1] "The Tiger and the Mouse" is an English story used for
[2] We are grateful to * * for the ìbìbìò translation Prosody investigation (rhythm and intonation).

Ekpe ye usịne ekesaña ke ikọt ekekịt eka isua uyo daña ana ke isọñ. Usịne ate, ekpe mbọk yak ami ndat uyo ami. ñkọ afo ukappa usuma uyo. Fọn ido yem ñkpọ mfen dia. Odo ekpe amaasio mbara ọmọ ọfʌk ke uyo ate, ami nnie. Muusọppọsọp idem udakka ke mmi, nya uta fiin ndian. Usịne amaadakka ye ekamba mfʌhọ.
Ekpe amaanwana edimen afịd uyo odo akañkeed odo uyo odo amaakefaha enye ke usʌñitọñ ndin domo daña enye ekpedodomo ikikanna ifakka.
Akekemkoo, ewa amaadiboiyo, ekpe abeeñe unwam. "Iniehe se ñkan nnam'', ewa ọbọọrọ asaña oboiyo. Akema mfuọt aditamma oboiyo, ekpe ebeeñe unwam. "Iniehe se ñkan nnam'', mfuọt ọbọọrọ atamma oboiyo.
Ke akpatre, ekpe amaamiañña adikisịm nte usịne ọdʌñ. Usịne mbon amaaketịppe isọñ ọdʌk asịne. "Mbọk nnyaña miin'', ekpe ate. "Uyo odo mmọdo amfaha ke usʌñitọñ, ndin ñkanna mfakka''. Usịne ate, "afo odo ata idiọk unam. Idəho ukuyakka nọ nta uyo odo, odo nyaasʌk unwam. Nwaññа inua mfo nọ ntamma ndʌk ñkwek uyo odo nọ asịp odo se aboiyo udʌk usʌñitọñ".
Ekpe amaanwaña inua, usịne ọtọọñọ edikwuek uyo odo. Ekpe ekere ete, ọbiọñ aanekke andọñ.

Figure 4. Processed text (output) from LAPTOL

Evaluating the LAPTOL we observe that out of 230 words processed:
Number of words correctly transliterated = about 65%
Number of words not tone-marked, but p2g translated (not in dictionary) = (15%)
Number of words neither p2g translated nor tone-marked (i.e. ignored and not in dictionary) = (21%)
Number of wrongly tone-marked words (underlined in figure 4) = (about 2%)

6. SUMMARY AND CONCLUSION

An effective G2P-P2G tasks deal with linguistics subproblems, some of which include parsing, sentence compression and word alignment and is applied to other fields by combining the solutions of these subproblems to end-to-end system such as TTs, information storage and retrieval, summarization, documentation etc., this combines a new algorithm for automatically creating a compound word lexicon from a text corpus with the G2P rule base. An inventory of character phone mapping has also been created on Ibibio text corpora. By using large text corpora from an input text sources, the system is able to extract and produces the phonetic transcription as may be required.

Essentially, high quality phonemisation for Ibibio can be achieved only by incorporating enough linguistic knowledge (about syllable boundaries, internal word boundaries etc.). G2P is a first step in this direction. Although it lacks some sources of knowledge (notably about sentence accent and syntactic structure), a transcription of high quality and accuracy can be obtained, and the system was successfully applied in practical tasks like rule testing and spelling error correction. The success of this work is focus towards the development of a well structured TTS system for Ibibio.

REFERENCES

[1] Allen, J., M.S. Hnnnicutt and D. Khttt (1987) From Text *to Speech.* Cambridge, UK: C.U.P.

[2] Berendsen, E., S. Language and H. van Leeuwen (1986) 'Computational Phonology: Merged, not Mixed.' Proceedings of COLING 86.

[3] Berendsen, E. and J. Don (1987) 'Morphology and stress in a rule based grapheme-to-phoneme conversion system for Dutch.' Proceedings European Conference on Speech Technology, Edinburgh.

[4] Berkel, B. Van and K. De Smedt (1988) 'Triphone analysis: A Combined Method for the Correction of Orthographical and Typographical Errors.' Proceedings 2nd ACt, Applied Conference.

[5] B. Narasimhan, R. Sproat, and G. Kiraz,(2011) Schwa dele-tion in hindi text-to speech synthesis,le in *Workshop on Computational Linguistics in South Asian Language*s.

[6] Chowdhury, G. (2003). Natural Language Processing. Annual Review of Information Science and Technology, 37: 51-89
http://www.cis.strath.ac.uk/research/publications/papers/strath_cis_publication_320.pdf

[7] Daelemans, W. (1985) 'GRAFON: An Object-oriented System for Automatic Grapheme to Phoneme Transliteration and Phonological Rule Testing.' Memo, University of Nijmegen.

[8] Daelemans, W. (1987) 'A Tool for the Automatic Creation, Extension and Updating of Lexical Knowledge Bases.' Proceedings of the Third ACL European Chapter Conference,

[9] Daelemans, W. (1987) *Studies in Language Technology: An Object- Oriented Computer Model of Morpho-phonological Aspects of GRAFON.* Doctoral Dissertation, University of Leuven,

[10] Daelemans, W. (1988) 'Automatic Hyphenation: Linguistics versus Engineering.' In: F. Stems and F.J. Heyvaert (Eds.), *Worlds behind Words,* forthcoming.

[11] Gibbon, D.; Urua, E.A.; Ekpenyong, M. (2004). Data Creation for ìbìbìò Speech Synthesis. LLSTI Third-Partners Workshop, Lisbon.
http://www.llsti.org/pubs/ ìbìbìò_data.pdf

[12] Jan P. H. van Santen and Adam L. Buchsbaum (1997), Methods for optimal text selection, in *Proc. Euro-speech '97*, Rhodes, Greece, pp. 553Œ556.

[13] Jeffrey D. Ullman Alfred V. Aho, John E. Hopcroft, *Data Structure and Algorithm*s, Addison-Wesley Publishing Company, Reading, MA, 1983.

[14] Kager, R. and H. Quen6 (1987) 'Deriving prosodic sentence structure without exhaustive syntactic analysis.' Proceedings European Conference on Speech Technology, Edinburgh.

[15] Kerkhoff, J., J. Wester and L. Boves (1984) 'A compiler for implementing the linguistic phase of a text-to-speech conversion system'. In: Bennis and Van Lessen Kloeke (eds), *Linguistics in the Netherlands,* p. 111-117.

[16] Lammens, J. M. G. (1987) 'A Lexicon-based Grapheme-to-phoneme Conversion System.' Proceedings European Conference on Speech Technology, Edinburgh.

[17] M. Choudhury and A. Basu,(2002), A Rule-based schwa deletion algorithm for Hindi,lh in *Proc. International Conference On Knowledge-Based Computer System*s, Navi Mumbai, pp. 343 Œ 353.

[18] Pounder, A. and M. Kommenda (1985). 'Morphological Analysis for a German Text-to-speech system.' COLING '86.

[19] Roger Tucker (2003), (Local language speech technology initiative, http://www.llsti.org)

[20] Ronald L. Rivest Thomas H. Cormen, Charles E. Leis-erson, (2000), *Introduction to Algorithm*s, Prentice-Hall of India, New Delhi.

[21] Sejnowski, TA. and C.R. Rosenberg. 'Parallel Networks that Learn to Pronounce English Text.' *Complex Systems 1,* 1987, 145 168.

[22] Stanfill, C. and D. Waltz. (1986) Toward Memory-based Reason- ing.' *Communications of the ACM,* 29 (12), 1213-1228.

[23] South African Keyboard User's Guide for the South African Keyboard and Fonts. (2006). http://www.translate.org.za.

[24] KỌNYIN Physical Multilingual Keyboards (2006) http://www.konyin.com/?page=home&menuitem=1

[25] Specification for a New Finnish Keyboard Layout. Final Draft. (2006). Keyboard Working Group of the Finnish National "Kotoistus" Initiative.

[26] Urua, E. (2000). Ìbìbìò Phonetics and Phonology. Centre for Advanced Studies of African Society, Cape Town.