Int J Speech Technol (2008) 11: 87–96
DOI 10.1007/s10772-009-9037-5

# Towards an unrestricted domain TTS system for African tone languages

**Moses E. Ekpenyong · Eno-Abasi Urua · Dafydd Gibbon**

Received: 18 June 2006 / Accepted: 10 July 2009 / Published online: 30 July 2009


**Abstract** In this paper we discuss the procedural problems, issues and challenges involved in developing a generic speech synthesizer for African tone languages. We base our development methodology on the "MultiSyn" unit-selection approach, supported by Festival Text-To-Speech (TTS) Toolkit for Ibibio, a Lower Cross subgroup of the (New) Benue-Congo language family widely spoken in the southeastern region of Nigeria. We present in a chronological order, the several levels of infrastructural and linguistic problems as well as challenges identified in the Local Language Speech Technology Initiative (LLSTI) during the development process (from the corpus preparation and refinement stage to the integration and synthesis stage). We provide solutions to most of these challenges and point to possible outlook for further refinement. The evaluation of the initial prototype shows that the synthesis system will be useful to non-literate communities and a wide spectrum of applications.

**Keywords** TTS · HLT · Multi-unit selection · Concatenative synthesis · Terraced tone modeling

M.E. Ekpenyong (✉) · E.-A. Urua
University of Uyo, PMB 1017, 520001 Uyo, Nigeria
e-mail: ekpenyong_moses@yahoo.com

E.-A. Urua
e-mail: anemandinyene@yahoo.com

D. Gibbon
Universität Bielefeld, Postfach 100131, 33501 Bielefeld, Germany
e-mail: gibbon@uni-bielefeld.de

## 1 Introduction

The goal of TTS synthesis is to convert arbitrary input text into intelligible and natural sounding speech using Human Language Technology (HLT). TTS methodology exploits acoustic representations of speech for synthesis, linguistic analysis of text to extract correct pronunciations and prosody in context. The evaluation of speech synthesis systems can be characterized into three folds: (i) accuracy of input text rendering; (ii) intelligibility of the resulting voice message; and (iii) the perceived naturalness of the resulting speech (Olive 1977). Today, TTS applications have been developed for information dissemination in various fields such as medicine, transport services, health, agriculture and weather information dissemination, and education in a wide range of subjects.

In the LLSTI project (Tucker and Shalonova 2005), the adaptation procedure was applied to Ibibio (ISO 693-2: nic; Ethnologue: IBB). To some extent, it may be practicable to adapt prosodically and phonemically similar languages, but adaptation complexity increases for typologically different languages (e.g. "intonation languages" for which TTS are typically developed, vs. "tone languages", e.g. Ibibio).

Speech synthesis development approaches in both academic and commercial fields have shifted to concatenative-based approaches (Mizuno et al. 2004; Hamza et al. 2005). They no longer attempt to derive an explicit model of speech production but expunge speech segments from a corpus of recorded speech, splice them together to produce synthesized utterances. Concatenation is generally believed to produce more natural and intelligible utterances than model based approaches, where the speech production process is parametrically modeled, and the model parameters varied in time to produce the speech.

The development of the Ibibio synthesizer involved a number of challenges related to the interface between speech technology and linguistics. For instance, (i) selecting appropriate phonetic units set; (ii) producing reliable pronunciations; and (iii) developing appropriate cost functions for selecting and joining diphone units.

## 2 Background issues

The development cycle of an experimental TTS prototype for a new language (from project design through resource acquisition to system design, implementation and evaluation) is rather a lengthy process, even when using older technologies such as MBROLA (Dutoit 1999) or modern technologies with unit selection shells such as Festival (Taylor et al. 1998) or Bonn Open Speech Synthesis (BOSS) system (Klabbers et al. 2001).

Gibbon et al. (2006) has analyzed in detail the linguistic and resource issues that impinge on Ibibio TTS system development. These issues are briefly summarized below:

1. *Linguistic issues: Language typology.* This includes key issues such as syllable phonotactics (a major determinant of the unit database resource), inflectional morphotactics (determining for instance whether table lookup or rule-based techniques are most appropriate for handling the vocabulary and its adaptation to grammatical context), morphotactics of word formation (tonal modifications in associative construction: context issues (as in úbÓk (hand) + Ńkáníká (clock/watch) = úbÓkŃkànìkà (the long/short hand of the clock/watch)) and sentence structure (combinatorics of words into units takes place with inflectional morphology).
2. *Resource issues.* This depends on the development environment in Nigeria and constitutes issues like human resources (training on both language documentation techniques and the Ibibio language), empirical resources (availability of experimental data) and infrastructural resources (stable electricity and Internet connectivity).

### 2.1 Tonal typology—summary

Tonal typology has also been effectively addressed by Gibbon et al. (2006) focusing attention on the consequences of language typological differences for TTS development in the present case on typology of tone (Gut and Gibbon 2002; Gibbon and Urua 2006). Explicit applications of Finite State Automata (FSA) to intonation modeling dates back to (Reich 1969), the 1970s IPO model ('t Hart and Cohen 1973; Pierrehumbert 1980) and (Gibbon et al. 1981). Many other intonation models, including (Hiroya et al. 1988) are also inherently FS models. Finite State Transducer (FST) modeling of tone-phonetics interface started with a modeling of two Niger-Congo languages (Baule, Kwa; Tem, Gur) in Gibbon (1987). The technique was extended to Mandarin tonal sandhi, which involved the mapping of lexical tone sequences or other lexical tone sequences (Martin 1998).

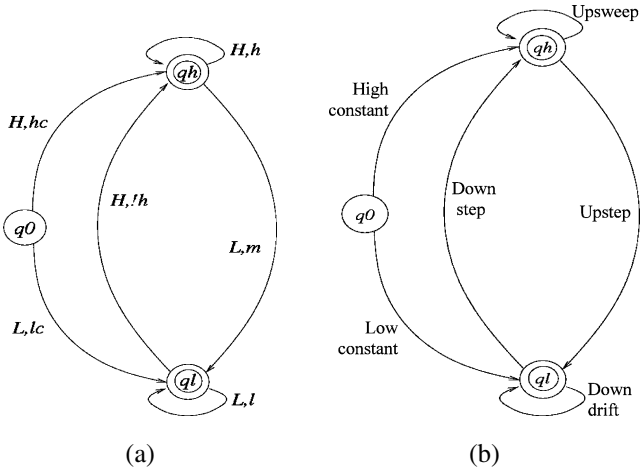



**Fig. 1** (**a**) Basic 2-tone Niger-Congo FST; (**b**) Generalisation of tone FST mapping types (Gibbon 2001). Key: H—High tone, L—Low tone, !—Down step, c—constant

Figures 1 and 2 show FSTs of the morphotonemic-phonetic interface (tone) typology for African tone system, in Gibbon et al. (2006).

The Western and Central African languages have a complex tonal structure and functionality when considering tone terracing vs. discrete level tone patterns, automatic and lexical downstep, upstep, downdrift and upsweep, tonal blocking, and tone-depressor consonants. The positional dependence of tone values on the 'terraced tone' patterning generated by automatic and non-automatic downstep in many African languages determines a further combinatorial explosion of pitch patterning (Gibbon 2001) in corpus markup and text pre-processing. On language modeling, the number of inflected word forms is far larger than for languages like English or Chinese due to the agglutinative inflectional morphology and complex subject-verb-object person concord in African languages, which adds further complexity to morphological tone assignment. This causes severe problems of text corpus sparseness as a specific case of the sparse data problem in empirical modeling.

However, in the area of tone representation for African tone languages, the following problems are most likely: terracing problems, words in context, homographs, loan words/personal names, numbers, dates and abbreviations. We are currently investigating these problems for Ibibio and formulating strategies to meet with the challenges that may likely arise.

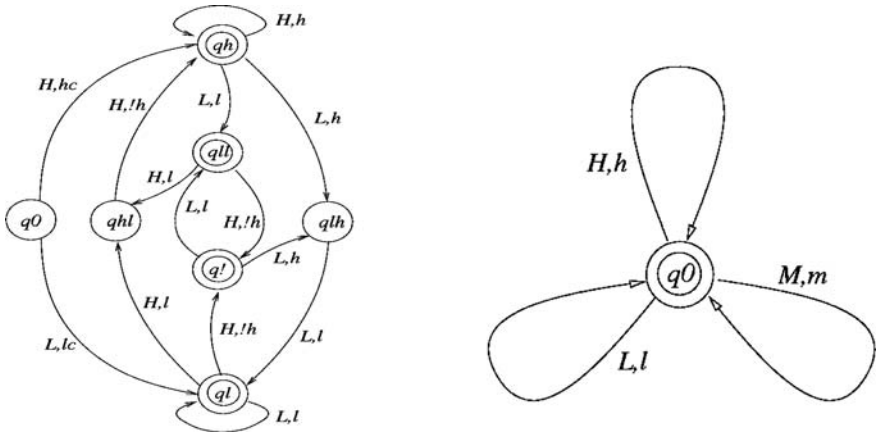



**Fig. 2** Variants: Baule FST (with lookahead) and 3-tone FST. Key: H—High tone, L—Low tone, !—Down step, c—constant

## 3 Ibibio synthesizer development

In this section, the procedure for developing a general-purpose TTS for Ibibio is presented. We also present from experience the possible problems that a developer should expect and propose solutions to the problems.

### 3.1 Corpus preparation and refinement

Corpus for a synthesizer could be collated from various sources (news readings, documentaries, textbooks, question and answer elicitation sessions, stories/fairy-tales, dictionaries, Internet, etc.). It is recommended that the corpus should be as large as possible in order to accommodate all the possible diphones of the language. During the text collection process, the text corpus must be validated before use.

*Determining a suitable representation for the orthography* For the Ibibio case, there was no electronic Ibibio corpus; consequently texts were collected and collated from different sources, typed, and checked for consistent orthography and font use. Texts that included special non-ASCII characters, such as accent markers or International Phonetic Alphabet (IPA) special symbols were normalized using character representations from the widely used Speech Assessment Methods Phonetic Alphabet (SAMPA) equivalent.

The problems encountered during the corpus preparation and refinement phase of our synthesizer development were:

i. Spelling mistakes/verbose sentences.
ii. Wrong phrase formation.
iii. Phrasal break symbol determination.
iv. Conversion of texts initially typed in orthographic form.
v. Time consumption.

Proposed solutions to the above problems include:

i. The flow of sentences and ease of reading should be taken into consideration and verified.
ii. All rules regarding the formation of phrases and sentences should be observed in order to ensure a well-formed input.
iii. Phrasal breaks should be marked with symbols that will not interfere with the programming language symbols.
iv. For orthographic texts, develop a Phoneme-To-Grapheme (P-2-G) parser. Consequently, a P-2-G parser (Gibbon et al. 2004) was developed as part of the research and used to convert orthography-based texts. In addition, the Ms Word and Open Office search and replace tools were also useful in the conversion process.
v. Encouragement (in the form of incentives) could be given to typing secretaries.

### 3.2 Selection of phonetically balanced sentences

The next step is to select sentences or phrases that are phonetically rich, i.e. contains all the diphones of the language. An open source text selection tool, Optimal Text Selection (OTS) for the selection of Phonetically Balanced Sentences has been developed at Hyberabad, India (Talikdar 2004). This tool requires scripts to generate the required OTS format. For Ibibio, a unix shell script was developed (Gibbon et al. 2004) to do the unit selection. A total of 165 phrases were selected from a corpus of about 4,500 phrases. The script is highly adaptable and can be used for other languages (it only requires the modification of the diphones conversion table).

### 3.3 Speaker selection and voice recording

When recording the Phonetically Balanced Sentences, the quality of the recordings is very important. The following should be considered as a minimum requirement:

1. A professional speaker with clear and consistent voice (that will not fade or change after a while) who has the ability of control volume and speed.
2. A good recording setup, preferably a recording studio that is acoustically damped.
3. Recorded sentences should be repeated at least three times (with a pause after each sentence) and the best of the three selected.

The recordings should be saved as wave (.wav) file format; this is the format compatible with Festival. The wave files are required in the annotation phase.

The following problems are likely to arise during this phase:

i. Recording mistakes.
ii. Cost of recording/acquiring quality equipment.
iii. Time consumption/loss of stamina.
iv. Choice of a speaker.

Below are proposed solutions:

i. The selected speaker should rehearse the sentences as many times as possible. A pre-test recording is recommended as preparatory measure. In our case, a professional and native Ibibio speaker with a stable prosody was chosen for the recording session.
ii. Funding could be sought for from local and international bodies to procure quality-recording equipment. For this research, a tripartite DAAD "Hochschulpartnerschaften" project with Universität Bielefeld, Germany; the University of Uyo, Nigeria and the Université de Cocody, Abidjan, Côte d'Ivoire provided travel scholarship to finance the cooperation, and the Humboldt Post Doctoral Fellowship has assisted in the procurement of quality recording equipment for the University of Uyo. The LLSTI project has also provided a one-year Industry scholarship and some equipment for the speech synthesis project. Though the recordings for this research were done in the acoustic laboratory of the University of Bielefeld, Germany to guarantee high quality recordings, a moderate setup is expected in the University of Uyo. The setup would result in a HLT group that will assist and encourage research students, staff and researchers alike to synthesize other African tone languages.
iii. The speaker should be physically, morally and emotionally sound and relaxed.
iv. Speakers should be encouraged and rewarded.

### 3.4 Speech annotation

The recordings could be annotated in any annotation software (for instance Praat, Transcriber, etc.). The tiers to annotate include the sentences, words, syllables, sounds and tones (both underlying and surface tones for tone integration). The text grid files from the annotations are necessary for generating the label (.lab) files in festival. A sample annotated file showing the spectrogram of one of the recordings is shown in Fig. 3.

The likely challenges encountered during annotations are as follows:

i. The need for proper training and expertise.
ii. Ensuring excellent results and accuracy.
iii. Time consumption.

The proposed solutions include:

i. Students should be trained to do the annotations as part of their continuous assessment (for instance when considering university projects) and the recordings should be checked later by experts before use.
ii. Patience and absolute carefulness are required during annotations; the annotations must be crosschecked for consistency.
iii. Encouragement (in the form of incentives) should be given to annotators to ensure absolute concentration and accuracy.

### 3.5 Voice integration and synthesis

#### *3.5.1 The speech synthesis engine*

In this section, we integrate an Ibibio female voice (a new voice) into the Festival Speech Synthesis System (Black and Taylor 1997) used as the synthesis engine for this research. In order to obtain state-of-the-art naturalness, a new unit selection approach known as MultiSyn (Clark et al. 2004) was used to select speech units to concatenate. In MultiSyn, a large text corpus is recorded in the same way as in the Classification and Regression Tree (CART) unit selection method (Hunt and Black 1996), but rather than use phones as a basic concatenation unit, diphones (the snippet of speech from the middle of one phone to the middle of the next phone) are used. This is because the middle of a phone tends more to be its acoustically most stable region. Therefore diphones represent acoustic transitions from the stable midsection of one phone to the next. A minimum of about 1,000 diphones is required to synthesize unrestricted English text (Schroeter 2006). The Ibibio language has about 625 diphones. With the tone-terracing problem, it is not yet clear at the moment the minimum number of diphones required for an unrestricted Ibibio tone synthesizer but we hope to achieve this in the course of this research. Since concatenative synthesis preserves the acoustic detail of natural speech, diphone synthesis is generally highly intelligible and produces superior synthesis quality when compared with other concatenative synthesis methods. The use of the diphone synthesizer in the Festival Speech Synthesis System is deprecated and

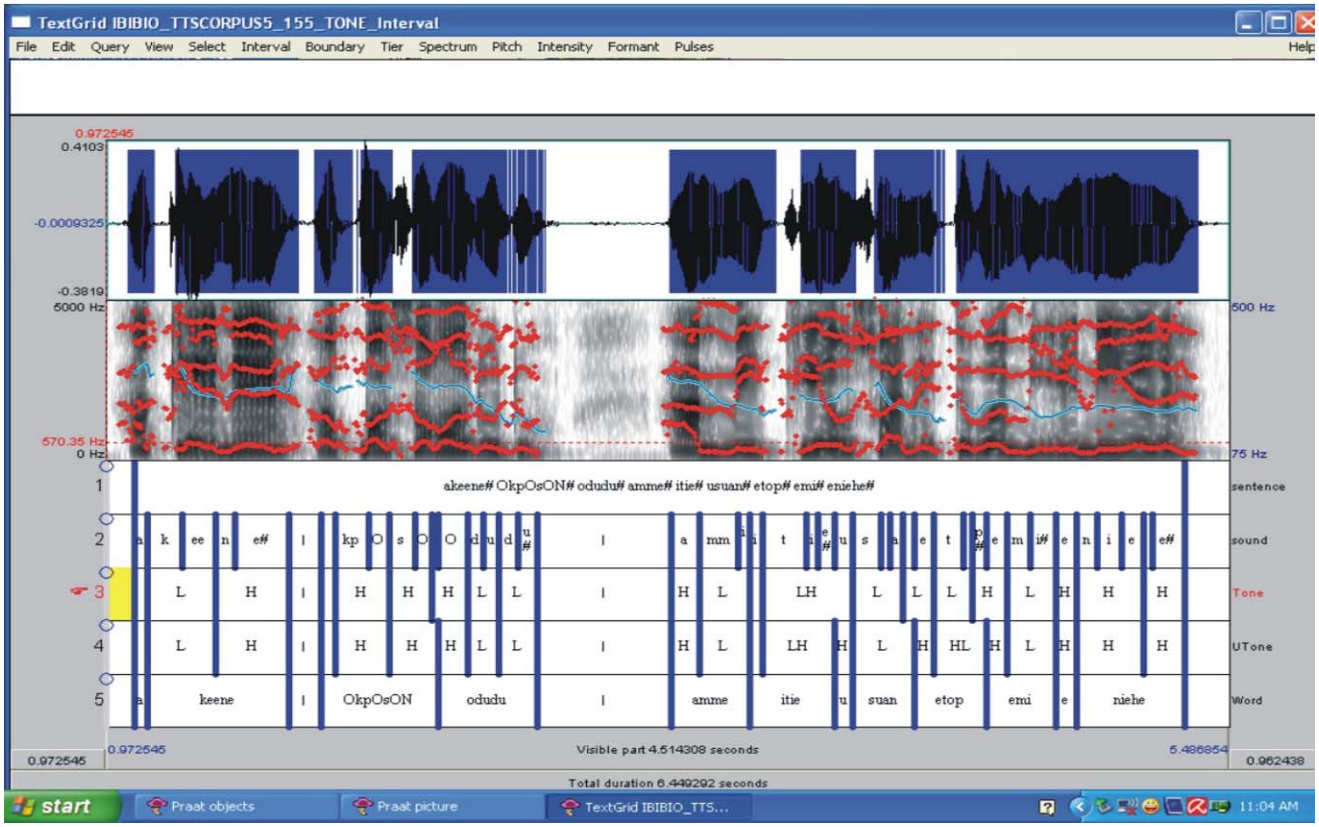



**Fig. 3** Sample annotation of a recording (including surface and underlying tone tiers)

could probably be removed from future versions. All of its functionality has been replaced by the UniSyn synthesizer (Black et al. 1999).

The advantages of MultiSyn synthesis over the more conventional cluster unit selection are that:

i. The target costs can be chosen to optimize the synthesis quality.
ii. All units are candidates for selection, whereas in the cluster unit selection, the search space for possible candidates is divided into clusters of acoustically similar units having same target costs; thus selected units may not be optimal in this case.

The disadvantages are related to its strengths: a large space must be searched, and target costs computed during synthesis, whereas with the cluster unit selection method, the target costs are inherently the clusters themselves, and the clustering process reduces the search space. This makes MultiSyn a bit slower.

### *3.5.2 Adaptation procedure: directory architecture*

We report in this section, the detailed steps followed during the pilot adaptation process of Ibibio into festival. These steps are strictly modifiable for other tone languages.

1. *Steps before the system compilation*

   i. Download and use the Snapshot_Version of Festival. The procedure below also works for recent versions.
   ii. Download Nina_voice unto the system and test it (this is used as an example of the directory structure for Ibibio voice). Nina_voice is a female voice. Any other voice could be downloaded and used.
   iii. Copy the Nina_voice folder and rename it as Eno_voice (Eno is the first name of the speaker). In this folder create a folder "eno" (this is not obligatory). The following directory structure should exist in the system:

   – /home/ekpenyong/Eno_voice/eno/utts.data—this is a required Festival file which contains all sentence prompts of the recordings—wave (.wav) files (in text form) for instance (ibibio_ttscorpus5_001 "bON akam kuukpa mba") (ibibio_ttscorpus5_002 "akefefeRe ajak ikOt abasi"). A Perl program (Gibbon et al. 2004) was written to automatically generate this file in Festival format.
   – /home/ekpenyong/Eno_voice/eno/lab/—should contain the label (.lab) files. A unix shell script (Gibbon et al. 2004) was written to automatically extract these data from the Praat text-grids. These

files were later checked and manually corrected for consistency. There should be a label file for a corresponding wave file.
- /home/ekpenyong/Eno_voice/eno/wav/—should contain the wave (.wav) files. The sample frequency of the wave files should be converted from 48000 stereo to 16000 mono, otherwise Festival will not recognize them. The unix regular expression: *sox filename001.wav –r 16000 – c 1 filename001-16k.wav rate* does the conversion.
- /home/ekpenyong/Eno_voice/eno/utt/—empty (will contain the utterance (.utt) files during compilation).
- /home/ekpenyong/Eno_voice/eno/etc/—empty.
- /home/ekpenyong/Eno_voice/eno/coef/—empty (will contain the duration coefficient (.coef) files during compilation).
- /home/ekpenyong/Eno_voice/eno/pm/—empty (will contain the pitch mark (.pm) files during compilation).
- /home/ekpenyong/Eno_voice/eno/lpc/—empty (will contain the linear pitch coefficient (.lpc) and the residual (.res) files during compilation).
- /home/ekpenyong/Eno_voice/eno/pauses/—has /coef, /pm, /utt and /wav sub directories and should be copied from Nina_voice without editing. The different sub directory files should be renamed, for instance, in our case . . . /pauses/coef has ibibio_ttscorpus5_001.coef, i.e. nina_x1_001. coef renamed).
- /home/ekpenyong/Eno_voice/eno/utts.pauses—should be copied from Nina_voice without editing.

iv. Create the folder iblex (for Ibibio lexicon)

- /home/ekpenyong/Festival_Snapshot_Version/festival/lib/dicts/iblex—the following scheme files should be copied into the folder: iblex.scm (ibibio lexicon file). This file defines the lexicon rules (i.e. Letter To Sound (LTS) rules and syllabification rules). And the ibibio_postlex.scm (ibibio post lexicon file), which defines the post lexicon rules. Both files were modified to include these rules for Ibibio.
- /home/ekpenyong/Festival_Snapshot_Version/lib/ibibio_phones.scm—Ibibio phone file (contains the phone inventory of Ibibio).

There are lexicons (scheme files) in other languages that could be modified or adapted to suit a desired language.

v. Copy build_unitsel.scm into /home/ekpenyong/Festival_Snapshot_Version/ cstr/scm/ and modify the path and file(s) calls.

vi. Create the folder . . . ibibio/uyo_eno_multisyn/festvox/.

vii. Copy uyo_eno_multisyn.scm into /home/ekpenyong/Festival_Snapshot_Version/festival/lib/voices_multisyn/ibibio/uyo_eno_multisyn/festvox/.

Note: In this file, the pathname of the user's PC must be changed, i.e. from "ekpenyong" to suit the user's system. The sentences can be synthesized using this file (uyo_eno_multisyn.scm).

2. *Compilation procedure*
The NOTES file in /home/ekpenyong/Festival_Snapshot_Version/cstr/ provides information on the compilation procedure. Importantly, FestVox must be installed on the user's PC as the compilation procedure uses some old functions from FestVox (/home/ekpenyong/Festvox/. . .). Other procedures are taken from the Snap Shot Version (/home/ekpenyong/Festival_Snapshot_Version/cstr/bin/).
Compilation could be done in four stages:

i. Generate Pitch marks: This is done by issuing the following commands at /home/moses/Eno_voice/eno> */home/moses/Festival_Snapshot_Version/cstr/bin/make_pm_wave pm wav/*.wav* */home/moses/Festival_Snapshot_Version/cstr/bin/make_pm_fix pm/*.pm*.
The pitchmark files will be generated into /home/moses/Eno_voice/pm.

ii. Generate Utterances: This is done by issuing the following command at /home/moses/Eno_voices/eno> *$FESTIVAL /home/moses/Festival_Snapshot_Version/cstr/scm/builds_unitsel.scm*.
On issuing the command, Festival will be loaded, at festival> issue:
*(build_utts "utts.data" /iblex)*.
The utterance files will be generated into /home/moses/Eno_voice/utt.

iii. Generate Normalized coefficients: This is done by issuing the command below at /home/moses/Eno_voice/eno> */home/moses/Festival_Snapshot_Version/cstr/bin/make_est_mfcc coef wav/*.wav*.

iv. Generate LPC coefficients: These files are generated by issuing the command below at /home/moses/Eno_voice/eno> */home/moses/festvox/src/general/make_lpc wav/*.wav*.

Note: These commands will function if the correct paths for the executables like: make_est_mfcc, make_lpc, etc. are specified. Use the *locate* command to know where they are installed.

*Utterance building* This process is one of the most difficult in voice compilation. During this phase, festival scans through the label file entries and matches each entry with each token stream in the utts.data file. The following files are required as input for utterance building.

– .scm files (both files and paths were described in 1.)
– .lab files
– .pm files
– utts.data (file with the orthographic representation of .lab files)
– build_unitsel.scm

To build the utterances, issue the commands:

– $FESTIVAL ../scm/build_unitsel.scm
– (build_utts "utts.data" 'iblex)

The output of the execution generates the .utt files (*.utt) into the /utt folder (/home/ekpenyong/Eno_voice/eno/utt/). Utterance building is carried out through synthesis, so any mismatches between the generated Letter To Sound (LTS) rules and transcription in the label files will cause errors. The only exception is that the pauses will be ignored in the label files if they do not correspond to the output of the phrase module ("phrase.scm").

*What can go wrong during utterance compilation?—Align mismatches issues*

i. A mistake in the annotation file (e.g. using a wrong symbol or using several symbols instead of one). Here a reference to the wave (.wav) files is required.
ii. Introducing more rules into ibibio_postlex.scm.
iii. Adding separate lexical items with a particular pronunciation (<name_of_the_lexicon>.scm).
iv. Introducing dummy phones (if other attempts fail).
v. Changing the code in build_unitsel.scm and introducing new possible mismatches as was done for pauses (this could be tricky).

For the Ibibio compilation, two errors were prominent: The "align mismatch" and "phoneset member" errors. The align mismatch error was debugged by modifying the syllabification of words in the label files (i.e. there were wrong syllabification of words during the generation of the label files). The corrections could also be made before utterance compilation. The phoneset member error which is as a result of missing phones and bad pitchmarking (phone clustering) was debugged by adding the flagged phone to /home/moses/Festival_Snapshot_Version/festival/lib/ibibio_phones.scm file. Bad pitchmarkings were automatically ignored by festival.

*Development of appropriate target-cost function* To select appropriate units during synthesis, a concatenative synthesis combines a target cost and a join cost. The target cost is the sum of the user-defined set of weighted components, each of which adds a penalty cost if some linguistic feature of the candidate diphone does not match the target, or if some default penalty feature is set in a candidate (which can be used to penalize candidates with poor labeling or bad pitch marking). The costs that could have significant influence on the quality of unit selection in tone languages are:

i. tone patterns matching
ii. word syllable positions
iii. number of syllables in word
iv. left and right contexts

3. *Running the system*
Once all the steps described above (with no compilation errors) have been carried out, the text to synthesize can then be tried. The following outlines the steps in running the system:

– Run Festival (change to the directory of festival, i.e. *cd. . . /. . . /festival*).
– At the prompt (festival>) type *(voice_uyo_eno_multisyn)*: this loads the Ibibio voice and generates the utterance.
– Then type *(SayText "<Ibibio Text>")*.

To save the output to wave file:

– At the prompt (festival>) type *(save_waves_during_tts)*.
– Then type *(tts_text "<Text to be said>" 0)*.

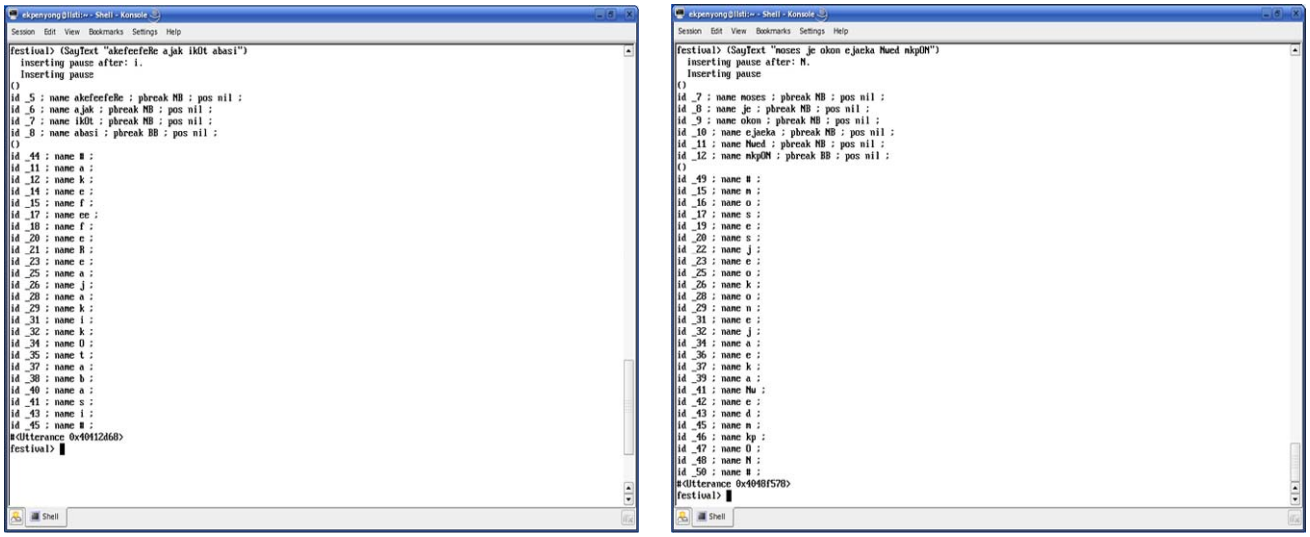
The output wave file will be named *tts_file. . . wav* in the current directory.

To create awareness, expand research interest and encourage collaboration, the Ibibio synthesis system is now available for download and trial. The compilation process has been accomplished and uploaded. The files are downloadable at http://llsti.org. For non-specialists wishing to try out the synthesizer but have difficulty downloading the archive, please contact the authors or any of the LLSTI members at http://llsti.org for guide. The LLSTI project has succeeded in publishing synthesizers for the following languages: (i) Hindi, an Indian language, (ii) isiZulu, a South African language, (iii) Kiswahili, a Kenyan language and (iv) Ibibio, a Nigerian language. The synthesizers are open-source and the scheme (.scm) files can be modified and adapted for the integration of other voices.

## 4 Pilot implementation

As a prototype to consider the feasibility of this research, the following tasks were accomplished, which require revisions and improvements:

1. Construction of the language data, e.g. electronic processing of the selected corpus and detailed checking.
2. Construction of the linguistic tools required for the TTS building, e.g. electronic lexical dictionary/lexical database based on Kaufman (1985) and Ibibio grammar (Urua 2000; Essien 1990).




(a) (b)

**Fig. 4** (**a**) Sample synthesis of the text "akefeefeRe ajak ikOt abasi". (**b**) Sample synthesis of the text "moses je okon eyaeka Nwed mkpON"

3. Construction of Ibibio diphone database.
4. Generation or extraction of the Phonetically Balanced Sentences using sentences (units) selection algorithm.
5. Recording of the Phonetically Balanced Sentences.
6. Annotation of the recorded sentences with the PRAAT phonetic workbench.
7. Integration of Ibibio grammar and Syllabification rules into FESTIVAL.
8. Compilation (unit selection and utterance building) in FESTIVAL.
9. Production of a prototype ("tone-deaf") Ibibio voice.

Further implementation issues are contained in Shalonova and Tucker (2004).

Sample synthesis of an Ibibio text with the prototype synthesizer is shown in Fig. 4.

Figure 4(a) shows a synthesis of an existing corpus text, while Fig. 4(b) is a formulated text with an English personal name "Moses" and an Ibibio personal name "Okon". The output of both sentences was properly syllabified and synthesized. The synthesis of the existing sentence (Fig. 4(a)) was smoother than the synthesis of a non-existing sentence (Fig. 4(b)), which though had some glitches, produced a synthesis with high intelligibility.

*Evaluation* Naturalness and Comprehensibility are the needed criteria for evaluating a TTS system. The subjective method (listener) ratings was used in this research to evaluate the TTS synthesizer.

The initial Ibibio TTS system was fully functional. An initial informal evaluation by native speakers (academic and non-academic staff of the Department of Linguistics and Nigerian Languages, University of Uyo) yielded encouraging results.

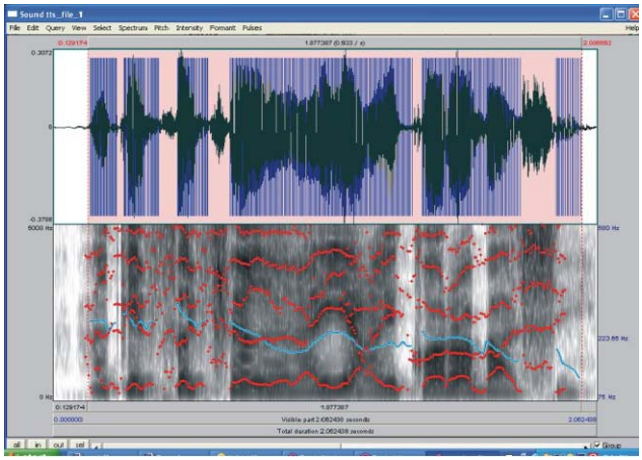
The LLSTI consortium in Shalonova and Tucker (2004) has presented the TTS development complexity score and a summary of the scores shows that Ibibio scores 5 for basic intelligibility and 7 for full intelligibility as opposed to Zulu which scores 6 and 8 respectively and other languages such as Hindi, Swahili and Tamil, which have lower scores. By basic intelligibility we mean generally correct G-2-P conversion and stress. Full intelligibility also has correct secondary stress, homograph disambiguation, etc.

In comparing the synthesis output with existing recordings, we observed that the synthesized sentence produced an average pitch of 223.65 Hz while the existing recording had an average pitch of 222.12 Hz, a negligible increase of 1.53 Hz. As mentioned earlier, a non-existing sentence was also synthesized with loan words (an English and Ibibio personal names). An average pitch of 223.86 Hz was obtained. The spectrograms for these sentences are shown in Fig. 5.

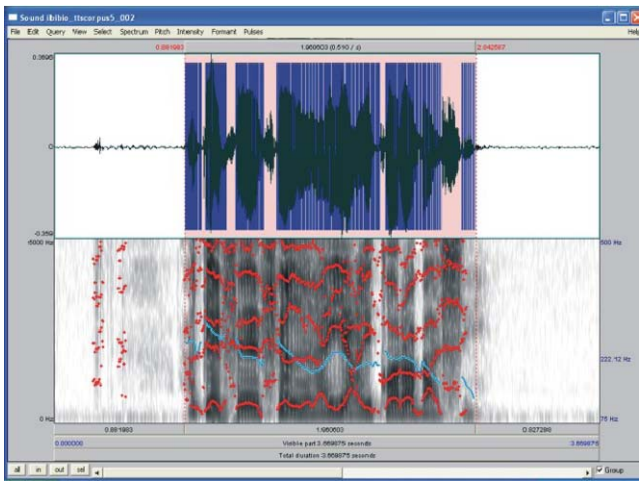
## 5 Future outlook and conclusion

The outcome of the pilot study (synthesis output) has been generally satisfactory with fair enough intelligibility and understandability. The next phase of the research will be the inclusion of tones into the synthesizer. The following are broad specifications of subsequent tasks:

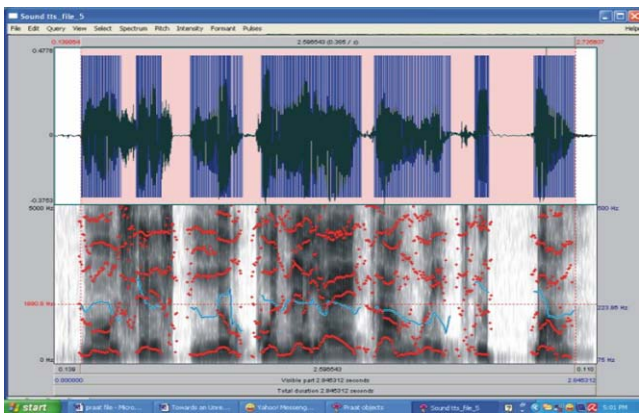
1. Annotation of tones in the dictionary, which has been completed.
2. Annotation of vowels in the corpus for vowel + tone (i.e. if there are $m$ vowels and $n$ tones, this will in principle yield $m * n$ such units), which initially may be enough to



(a)



(b)



(c)

**Fig. 5** (**a**) A spectrogram of a synthesized text (existing) "akefeefeRe ajak ikOt abasi". (**b**) A spectrogram of an existing (recorded) text "akefeefeRe ajak ikOt abasi". (**c**) A spectrogram of a synthesized text (non-existing) "moses je okon ejaeka Nwed mkpON"

annotate with the lexical tones from the dictionary. Currently the present 165 extracted sentences have been re-annotated with tones.

3. Resulting annotations can then be filtered automatically, for instance with a Finite State Transducer script (Gibbon et al. 2004) in order to index each vowel + tone unit with its sequential position in the sequence. This will yield vowel + tone + position annotations.
4. The algorithm for selecting the phonetically richest sentences can then be rerun (and re-defined to some extent).

In the area of generation/synthesis, the inputs will need to be enriched with tones. This is far from trivial. There are several problems to cater for, which include homographs/homophones with different tones and different meanings, morphological tones (e.g. distal/proximal future).

We wish to investigate the number of diphones captured by the pilot annotations (the 165 Phonetically Balanced Sentences). Thus, by developing a Perl/Unix script to scan the annotation files for existing diphones, we can determine missing diphones and formulate a handful of corpus to compensate for the missing diphones and frequently occurring loan words. This reverse engineering approach will, to a great extent, solve the problem of gathering additional large corpus for re-selection purposes.

The issue of tone terracing is also being looked at. A computational scheme/implementation that would be adaptable to other tone languages is expected. In the long run a Web evaluation of the synthesizer will be made. This idea will greatly improve the synthesis quality in terms of intelligibility.

Though more funding is required to complete this huge and challenging research, it is hoped that with the available contacts and current research results, a generic African tone language synthesizer will evolve.

**Acknowledgements** We wish to acknowledge Roger Tucker and Ksenia Shalonova of the Outside Echo, UK for the one-year industry research scholarship that gave birth to the Ibibio LLSTI project and this paper. LLSTI also provided the working equipment and training on speech engineering and the festival synthesis system. Prof. Dr. Dafydd Gibbon of Universität Bielefeld, Germany cannot be left out for his untiring support to the project and for providing the excellent research/working environment.

## References

Black, A., & Taylor, P. (1997). *Festival speech synthesis system: system documentation (1.1.1)*. Human Communication Research Centre, Technical report. HCRC/TR-83.

Black, A., Taylor, P., & Caley, R. (1999). *The festival speech synthesis system*. System Documentation (1.4.0), www.cstr.ed.ac.uk/projects/festival/manual/.

Clark, R., Richmond, K., & King, S. (2004). Festival 2: build your own general purpose unit selection speech synthesizer. In *5th ISCA speech synthesis work shop*, Pittsburgh, PA (pp. 173–178).

Dutoit, T. (1999). *An introduction to text-to-speech synthesis*. Berlin: Springer.

Essien, O. (1990). *A grammar of the Ibibio language*. Ibadan: University Press Limited.

Gibbon, D. (1981). A new look at intonation syntax and semantics. In A. James & P. Westney (Eds.), *New linguistics impulses in foreign language teaching*. Tübingen: Gunter Narr

Gibbon, D. (1987). Finite state processing of tone systems. In *Proceedings of the European chapter of ACL*, Copenhagen (pp. 291–297).

Gibbon, D. (2001). Finite state prosodic analysis of African corpus resources. In *7th EUROSPEECH conference*, Aalborg, Denmark (pp. 83–86).

Gibbon, D., & Urua, E. (2006). Computational morphotonology in Niger-Congo languages. In *Proceedings of speech prosody 2006*, Dresden, Germany.

Gibbon, D., Urua, E., & Ekpenyong, M. (2004). *Data creation for Ibibio speech synthesis*. LLSTI Progress Report, Third Partners Workshop, Lisbon.

Gibbon, D., Urua, E.-A., & Ekpenyong, M. (2006). Problems and solutions in African tone language text-to-speech. In *MULTILING 2006 ISCA Tutorial and Research Workshop (ITRW)*, Stallenbosch, South Africa.

Gut, U., & Gibbon, D. (Eds.) (2002). *Typology of African prosodic systems*. Bielefeld occasional papers on typology 1. Universitaet Bielefeld, Germany.

Hamza, W., Bakis, R., Shuang, Z., & Zen, H. (2005). On building a concatenative speech synthesis system for blizzard challenge speech databases. In *INTERSPEECH 2005*, Lisbon.

Hiroya, F. (1988). A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour. In O. Fugimura (Ed.), *Vocal physiology: voice production, mechanisms and functions* (pp. 347–355). New York: Raven Press.

Hunt, A., & Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of ICASSP, 1*, Atlanta, Georgia (pp. 373–376).

Kaufman, E. (1985). *Ibibio dictionary*. Cross River State University and Ibibio Language Board, Nigeria, in cooperation with African Studies Centre, Leiden, The Netherlands.

Klabbers, E., Stoeber, K., Veldhuis, R., & Breuer, S. (2001). Speech synthesis development made easy: the Bonn open synthesis system. In *Proceedings of Eurospeech*, Aalborg (pp. 521–524).

Martin, J. (1998). A two-level take on Tianjin tone. In G.-J. Kruijff & I. Kruijff-Korbayová (Eds.), *Proceedings of the third ESSLLI student session, 10th European summer school on logic, language and information*, Saarbruecken, Germany (pp. 162–174).

Mizuno, H., Asano, H., Isoyai, M., Hasebe, M., & Abe, M. (2004). *Text-to-speech synthesis technology using corpus-based approach*. NTT Technical Review (Vol. 2, No. 3, pp. 70–75).

Olive, J. (1977). Rule synthesis of speech from diadic units. In *Proceedings of ICASSP-77* (pp. 568–570).

Pierrehumbert, J. (1980). *The phonology and phonetics of English intonation*. Diss. Massachusetts Institute of Technology.

Reich, P. (1969). The finiteness of natural language. *Language*, *45*, 831–843.

Schroeter, J. (2006). Text-to-speech (TTS) synthesis. In R. Dorf (Ed.), *Circuits, signals and speech and language processing*. http://www.research.att.com/~ttsweb/tts/papers/2005_EEHandbook/tts.pdf.

Shalonova, K., & Tucker, R. (2004). Issues in porting TTS to minority languages. In *SALTMIL workshop on minority languages, LREC 2004*, Lisbon.

Talikdar, P. (2004). *Optimal text selection module version 0.2*. LLSTI Progress Report, Third Partners Workshop, Lisbon.

Taylor, P., Black, A., & Caley, R. (1998). The architecture of the festival speech synthesis system. In *3rd ESCA workshop on speech synthesis* (pp. 147–151), Jenolan Caves, Australia.

't Hart, J., & Cohen, A. (1973). Intonation by rule, a perceptual quest. *Journal of Phonetics*, *1*, 309–327.

Tucker, R., & Shalonova, K. (2005). Supporting the creation of TTS for local language voice information systems. In *INTERSPEECH-2005* (pp. 453–456).

Urua, E. (2000). *Ibibio phonetics and phonology*. Cape Town: Centre for Advanced Studies of African Society.