Supporting the Creation of TTS for Local Language Voice Information Systems

Roger Tucker and Ksenia Shalonova

Outside Echo Ltd, Chepstow, UK
roger,ksenia@outsideecho.com

Abstract

We report on the Local Language Speech Technology Initiative, which is producing the TTS required for voice information systems in the developing world. We overview the whole process now the initial phases of Hindi, isiZulu, Kiswahili and Ibibio are complete, outline some applications we are targeting, and draw some lessons for the future.

1. Introduction

In the developing world, where the vast majority of the world's population live, mobile phone usage is growing at a phenomenal rate. Many users would value timely information on jobs, health issues, local market prices etc. but have little access to computers and know only their local or national language. Voice services & speech technology would address this need but have to be adapted both linguistically and culturally with different service models in order to succeed for these new users, who represent a completely different and new market for voice systems.

The most fundamental adaptation needed is to create the raw technologies – TTS and ASR - in the local languages. This is not an attractive proposition for speech companies, with an unproven market, awkward language and script issues to overcome (African tone for example), and a general lack of good resources. In contrast, available development packages like Festival[1] and MBROLA[2] are frequently downloaded by motivated individuals and used to build basic TTS systems in their own languages. These rarely have the high quality needed for voice information systems, the problem being that good quality TTS requires extensive expertise and experience to produce.

Conceived during Eurospeech 2001, the Local Language Speech Technology Initiative (LLSTI) has been formed to try a new approach to the production of speech technology in local languages particularly in the developing world. LLSTI provides training and on-going support to enable a non-expert to produce a good quality system in a reasonable time frame, with the resulting system going open source so that others can build on it, thereby creating an accessible community of interest in both the language and the technology.

The initial phase has lasted a little over 15 months and set out to support four language partners developing TTS and another one extending the toolset to allow for morphological analysis. Some of the partners have given their time and IP without charge, others have required modest levels of funding which we have obtained from various sponsors (see Acknowledgements).

Previously we reported on some of the issues that were arising during the work on the first four languages – Hindi, Kiswahili, isiZulu and Ibibio [1]. In this paper we overview the whole process now the initial phase is complete, and draw some lessons for the future for supporting the production of speech & language technology remotely.

2. Overview of TTS development

2.1. Aims

This initial phase set out to produce basic systems consisting of these modules:

- sentence extraction (including differentiation of main intonation patterns – questions and statements)
- tokenisation (word extraction)
- stress/tone assignment[3]
- morphological analysis (MA) and POS tagging. This is required in the basic system only if morphological-syntactic characteristics in a particular language are needed to obtain the correct phoneme sequence.
- phrase extraction based on punctuation marks etc.

Although we had planned to, we were not able in the end to include MA for any of the languages mainly because of the tight timescales involved (see Section 2.3).

A full system would also include these modules, on which good progress has been made for most of our languages:

- text normalization (acronyms processing, abbreviations processing, special symbols processing, digits processing)
- loan words processing
- proper names processing
- phrase extraction based on morpho-syntactic information when there are no punctuation marks.

We have used Festival 1.95, using the Multisyn unit selection engine from CSTR Edinburgh [2]. Unit selection offers very natural synthesis, an important requirement for voice services.

2.2. General Procedure

Our four languages (Hindi, Ibibio, Kiswahili and isiZulu) have all followed the same development procedure. Additional

[1] http://www.cstr.ed.ac.uk/projects/festival/

[2] http://tcts.fpms.ac.be/synthesis/mbrola.html

[3] Tone assignment has to be fulfilled on this level if the tones distinguish lexical/grammatical meaning.

resources or modules created for each particular language are described in section 2.4.

1. *Speaker selection.* The definition of a normative speaker is not straight forward for some languages and is described in [1]. Our most successful voice is the Kiswahili voice of Ken Walibora, a national TV anchorman, renowned Kiswahili novelist and master of the standard Kiswahili dialect Kiugunja. We are as convinced as ever that the choice of voice is the single most important decision in the whole system.

2. *Insertion of language-dependent data* into the following Festival modules:

    - Phoneset
    - G2P rules
    - Syllabification rules
    - Tokenisation
    - Phrasing rules (based on punctuation marks)
    - POS tagger
    - Text Normalisation module - completed only for Kiswahili.

    An alternative G2P module (that can work both within Festival and independently) was produced in C++ by HP Labs India [3]. This module allows deletions, insertions and replacements of sound combinations due to the context described by a user. It can also be used in the process of creating Phonetically Balanced Sentences for phonetically transcribing large text corpora.

3. *Phonetically Balanced Sentences* were produced using a Greedy algorithm based tool also produced by HP labs India [4]. One of the options of this tool is the ability to use diphone or syllable frequency characteristics, i.e. to select sentences containing more representatives for most frequent speech units. The program uses the following *input* files:

    - full set of allophones/phones with their frequencies.
    - text corpus in orthographic version
    - phonetic transcription of this corpus.

    And produces these *output* files:

    - selected sentences in orthographic form.
    - Non-covered units - i.e. the units that were not found in the transcribed corpus by the Greedy algorithm (these units can be either impossible sound combinations or infrequent ones).

4. *Recordings* made of the phonetically balanced sentences. Only declarative sentences were recorded for the initial phase.

5. *Annotation* of the recordings. The current Multisyn unit selection algorithm does not allow modification of prosodic features, so the more detailed the database annotation, the better the intonation.

6. *Compilation* of the whole system in Festival. This proved the most difficult technical part that required from the developer experience in Linux and scripting languages. We provided simple documentation and a bash script (with main commands) for people not experienced in Linux. The existing Multisyn tool for assigning join coefficients (used in join cost for unit selection algorithm) was improved by CSIR, Pretoria.

7. *Testing* (see section 3)

2.3. Morphological Analysis

For three of the languages MA is essential for providing proper transcription, with different degree of importance - for Hindi schwa deletion rules require MA, for Ibibio and isiZulu tone assignment requires MA. For all four languages MA is required for prediction of phrase boundaries when there are no punctuation marks.

LLSTI partners IIIT Hyderabad have designed and built a new tool which can perform MA for almost all languages by filling in data tables [5]. Unfortunately, it was not ready in time for use in this initial phase, so all our languages have had to do without MA. This tool, which uses a context free mechanism, can be easily adapted for TTS modules: G2P, phrasing, stress and tone assignment and POS tagger. The tool is available standalone as well as a Festival module.

2.4. Specific Language Work

The LLSTI project began with a survey of 105 languages to identify all the script and language features which can create complications for a TTS system. We catalogued their TTS-related features in our TTS-Related Multilingual Database [6][1]. This enabled us to predict the issues that specific languages would face, and we have summarized these in a TTS development complexity score as shown in Table 1.

Table 1: Complexity Scores for the four languages

Language	Basic	Full
Kiswahili	0	5
Hindi	2/3	4/5
Ibibio	5	7
IsiZulu	6	8

2.4.1. Kiswahili

The Kiswahili work was carried out by the University of Nairobi. Kiswahili is the simplest language in our set - it has direct g2p rules, no tones and fixed stress. The morphological information in Kiswahili is required only for predicting phrase boundaries. To generate good intonation, the speech database was annotated differentiating three types of stress: vowels under lexical stress, vowels under phrasal and sentence stress and non-stressed vowels. The database uses just 414 declarative sentences [7] yet covers all contexts and generates TTS of good quality. The professional speaker has also made a big difference to the naturalness of the Kiswahili TTS system.

[1] see http://www.llsti.org/languages-database.htm

2.4.2. Hindi

The Hindi work was cariied out by HP Labs Bangalore, India. Hindi has got almost direct g2p rules except for the phenomenon of schwa deletion. This was partly solved by using extended g2p rules, but to completely predict schwa deletion, MA would be required.

The place of the lexical stress in Hindi is a disputable question and was disregarded for this initial phase. Hindi is the only one of our initial four languages written in a non-latin (Devangari) script. A special converter was developed for transliterating Devangari script to latin characters. This converter is a part of the general language-independent g2p system mentioned in section 2.2.

2.4.3. Ibibio

The most difficult language from the TTS development point of view was Ibibio, which was carried out by the Univeristy of Uyo, Nigeria with substantial support from the University of Bielefeld, Germany. Although its complexity score is very close to that of isiZulu, lack of any resources and in particular lack of written texts required a lot of preliminary work to be done before TTS development.

Ibibio orthography does not contain tone marks and there are three steps for tone assignment: lexical, morpho-syntactic and terraced tones that require morpho-syntactic analysis based on agglutinating morphology. FST rules are required for modelling terraced tones [8]. Whilst these problems were predicted from the TTS-related multilingual database, we didn't fully anticipate the almost complete lack of available written texts in Ibibio. Also there are many loan words due to Nigeria's colonial past which will need to be dealt with in the full system.

Although the current Ibibio TTS system is a very basic one, the amount of preparation work (creation of text corpus and dictionaries, finite state model for grammatical and terraced tones) is of great value. These resources will be used for improving the current system.

2.4.4. isiZulu

The isiZulu work was carried out by CSIR, Pretoria, South Africa. isiZulu orthography does not contain tone marks and has the same three steps for tone assignment as Ibibio. The current isiZulu system is "tone deaf" - i.e. tones are used randomly as they are not distinguished in the unit selection process - but tests suggest that this misuse of tone is nowhere near as detrimental for isiZulu as for Ibibio. This may be because isiZulu speakers have become accustomed to hearing incorrect tone usage by English/Afrikans speakers.

CSIR Pretoria are developing techniques for dealing with the lack of resources for languages like isiZulu. Their g2p was obtained with a bootstrapping approach which iteratively creates pronunciation dictionaries and then derives the rules from them [9]. They also have been developing isiZulu MA using a bootstrapping framework. For producing intonation semi-automatically, they have created a MOMEL/INTSINT intonation module for Festival[1] which has been tested on isiZulu data with promising results [10].

3. Testing & evaluation

The testing of TTS systems is complicated by the subjective nature of the whole process. So far testing has been a little ad-hoc using one or more of these approaches:

1. Developer testing (testing intelligibility). High-level testing to eliminate major bugs in all text processing modules such as MA, g2p, phrasing and also checking the diphone coverage.
2. User tests (testing naturalness and intelligibility). Tests that are given to native subjects asking them for vague comments (e.g., "the word has got the wrong stress", "the intonation is unnatural", "the sound is too artificial" etc.). Although linguists could be asked to carry out such tests, this can give a rather prejudiced opinion.
3. Real Application testing (testing naturalness and intelligibility in connection with social factors). Testing TTS systems in real life applications is important as it is the goal of our TTS systems. For instance, the Hindi system has been tested in a system for booking railway reservations and the isiZulu system has been tested in an information delivery system for weather, health and tax[11].

In the future we aim to test all our languages with all three approaches.

One interesting aspect of the isiZulu testing was the results from literate users vs illiterate users. There was no significant difference between the two groups in the subjective assessment of either naturalness or intelligibility. But in a comprehension test, the illiterate subjects showed considerable confusion about the information given. This underlines the need for adapting the systems for the needs of the particular user group they are targeted at.

4. Local Language Applications

In this section we outline some of the voice services and other applications we are targeting.

4.1. Phone-based Services

LLSTI partner CSIR's OSISA-sponsored HIV/AIDS Information Service (currently under development) is a good example of a telephone information service provided for a *specific user group*. It provides basic information (i.e. what to do, who to contact, where the nearest support service is located etc) from a database – currently such information is often patchy, incomplete, outdated or inaccessible to those who need it. But the service also links into the existing telephone counselling services, so a real person can be made available when needed. It serves both HIV sufferers and people providing home care.

People groups who would particularly benefit from such specific phone information services are:

[1] Available from http://www.llsti.org/downloads-tools.htm

- Young women, particularly in rural areas – an information and networking system can provide information on training, employment services and health.
- Farmers – market prices, veterinary advice, relevant warnings etc. The services can target particular farming communities, addressing issues specific to those communities.

We are also looking at voice versions of information services currently provided by SMS messaging. Voice allows an SMS service to be extended beyond 160 characters. But there is a significant mobile user population, associated with low economic status, who never use SMS at all. A good example of a service which would offer such users real economic value is the *560* job brokering service run by Oneworld in Kenya. Here tradespeople can register their trade (e.g. plumber, painter) and prospective employers can send job vacancies which are then distributed to all registered.

4.2. PC based usage

Kiosk sevices - in countries like India, kiosks are mostly manned. The kiosk operator provides the interface to services such as health, e-governance, agriculture and education (see http://www.n-logue.com/services.htm). Projects like eNRICH (see http://www.enrich.nic.in) are producing kiosk software to make them more accessible to non-computer users. Spoken output along with such simplified user interfaces would allow direct access for everyone, allowing more privacy and individual freedom in computer use.

Screenreaders - ironically, even in countries where one of the major languages are taught in schools, visually impaired people have least access to the school system and therefore tend to know only their local languages. However, producing a fast and interactive screenreader will require further work, as Multisyn is not designed for speed.

5. Discussion

The four languages have very different TTS-related complexity and given that they have been developed in the same timeframe it would be unfair to compare their quality at this stage. What we *have* shown is that with no prior experience in speech technology it is possible for a LLSTI partner to produce a good quality TTS system in a short period of time (6-8 months) for a “simple” language like Kiswahili.

At present all the language teams consist of engineers with linguist support. Over time we would like to increase the amount of work which could be undertaken directly by the linguists by abstracting out the data required by each module. This is the philosophy we have adopted with the new MA module.

Lack of internet connection (and often power) in Nigeria severely affected the Ibibio work. Although we had to some extent planned for this by arranging two extended visits to Europe, it did prove very difficult to support the work once the engineer concerned had returned to Nigeria. We have learnt that the training needs to be a lot more comprehensive in these conditions.

6. Conclusion

The initial phase of LLSTi has shown the viability of a partnership approach to local language TTS development. The project is continuing with more languages (Nepalese, Setswana) and working towards the deployment of the existing languages in both voice service and screenreading applications.

7. Acknowledgements

LLSTI is grateful for the sponsorship of the Department for International Development (DfiD), UK, the International Development Research Centre (IDRC), Canada, and Oneworld International, UK.

We would like to thank all our partners for their contributions to the success of the initial phase: Prof. Dafydd Gibbon - University of Bielefeld (Germany); Prof. Ramakrishnan - IISc Bangalore (India), Kalika Bali – HP Labs (India), Prof. Etienne Barnard, Marelie Davel and Aby Louw - CSIR (South Africa), Prof Eno-Abasi Urua and Moses Ekpenyong - University of Uyo (Nigeria); Dr. Mucemi Gakuru – University of Nairobi (Kenya).

8. References

[1] Shalonova, K. and Tucker, R., "Issues in Porting TTS to Minority Languages", *SALTMIL workshop on Minority Languages, LREC 2004, Lisbon, pp 76-79.*

[2] Clark, R.A.J., Richmond, K. and King, S., "Festival 2 – Build your own General Purpose Unit Selection Speech Synthesizer", *5th ISCA Speech Synthesis Workshop, Pittsburgh, June 2004 , pp173-178.*

[3] Bali, K. et al, "Tools for the Development of a Hindi Speech Synthesis System", *5th ISCA Speech Synthesis Workshop, Pittsburgh, June 2004*

[4] Talukdar, P., "Optimal Text Selection Module V0.2", *available from* http://www.llsti.org/downloads-tools.htm

[5] Sangal, R., Sharma, D.M. and Mamidi R., "Generic Morphological Analysis Shell", *SALTMIL workshop on Minority Languages, LREC 2004, Lisbon, pp 40-43*.

[6] Shalonova, K. and Tucker, R., "South Asian Languages in Multilingual TTS-related Database", *EACL Workshop on Comp.Ling. for South Asia, Budapest,2003, pp. 57-63*

[7] Gakuru, M. et al, "Design of Speech Data Base for Unit-Selection in Kiswahili TTS", *E-Tech2004, Kenya*

[8] Gibbon, D. "Tone and timing: two problems and two methods for prosodic typology", *TAL-2004 ISCA Symposium on Tonal Aspects of Languages, Beijing, China, March 28-31, 2004, pp 65-72*

[9] Davel, M. and Barnard, E., "The Efficient Generation of Pronunciation Dictionaries: Machine learning Factors during Bootstrapping", *ICSLP 2004, Korea*

[10] Louw, J.A. and Barnard, E., "Automatic intonation modeling with INTSINT", *Proc 15th Annual Symp. of Pattern Recognition Association of S. Africa, Grabouw, Nov 2004 pp. 107-111*

[11] Barnard, E. and Davel, M., "LLSTI isiZulu TTS Evaluation Report", *available from* http://www.llsti.org/pubs/Zulu_testing.pdf