Grapheme Based Tigrinya Speech Synthesis using Statistical Parametric Speech Synthesis

Luel Negasi
HiLCoE, Computer Science Programme, Ethiopia
luelnegasi@gmail.com

Wondowssen Mulugeta
HiLCoE, Ethiopia
School of Information Science, Addis Ababa University, Ethiopia
wondwossen.mulugeta@aau.edu.et

Abstract

In this paper a model-based speech synthesis prototype for Tigrinya language idiom is developed in an integrated speech synthesis framework (Festival speech synthesis system).While the frontend of the framework is Grapheme based synthesizer, the backend is CLUSTERGEN Synthesizer which is an instance of statistical parametric speech synthesis. The under resourced linguistic nature of the language was the main reason to choose this framework. 249 Tigrinya graphemes were considered as phonemes independently; irrespective of its 32 phonological phonemes. For this study, 800 previously prepared sentences and rerecorded again in a recommended way is used as corpus. Amendments and additions to the adopted methodology was done. The whole prototype synthesis development was done automatically. A tenfold threshold method was used for training and testing of the prototype. The synthesized speech was Android deployable prototype. This synthesized speech resulted in a score of 5.82 using Mel Cepstral Distortion (which is built-in objective measurement metric); while subjective evaluation resulted in 4.5 and 4.3 out of 5 score, naturalness and intelligibility of the synthesized speech respectively. Both evaluations were interpreted as the synthesized speech was almost the same as natural human speech.

Keywords: Speech Synthesis; Statistical Speech Synthesis; Grapheme; Spoken Language

1. Introduction

Speech synthesis is playing different roles in today's modern human activities like aid for the disabled and in the telecommunication sector [11]. Its task is converting written text to speech. Naturalness and intelligibility are the main quality measurements of a synthesized speech. Meanwhile text to speech consists two main sub parts known as natural language processing and digital processing that are also termed as frontend and backend systems respectively [11].

Frontend systems are used for processing linguistic part of a language that is going to be synthesized. Many of its tasks relay on the nature of a language. In general, it consists of activities such as text processing, converting text to sound and prosody modeling.

In the past few decades, different synthesizing methods (backends) have been implemented. Nowadays the popular method is statistical parametric speech synthesis which is model based [4, 11, 16].

Tigrinya is a Semitic language spoken mainly in Eritrea and northern part of Ethiopia (Tigray). It is phonetic to mean that what is written is what it pronounces [6, 12, 13]. Meanwhile it is under resourced language in its linguistic part that can be used as an input for speech synthesis mainly for the frontend part.

Very few attempts in synthesizing Tigrinya had been done such as those by Tesfay Yihdego [15], Agazi Kiflu [7] and Lemlem Hagos [5]. All these attempts are based on different types of concatenative synthesis techniques. They used different transliteration mechanisms but all were tried

to transliterate the Unicode based orthography to Latin alphabets in order to get pronunciation dictionaries using these Latin alphabets. This way of finding pronunciation dictionaries adds extra process though it is possible to convert its Unicode letters directly to International Phonetic Alphabet (IPA). Moreover, all attempts were based mainly on the phonological phonemes of the language. In addition to that the backends they used were not model based techniques. The frontend and backend systems used to synthesize Tigrinya language so far lack integrity.

In this paper 249 graphemes of the language were used as means of finding pronunciation. This is because it is easy if the graphemes are considered as phonemes since Tigrinya writing system is phonetic. Grapheme based Synthesizer which is integrated in Festival/Festvox framework was used as frontend.

The backend system used was also model based statistical parametric speech synthesis method known as CLUSTERGEN Synthesizer.

1.1 Tigrinya Language and its Writing System

Tigrigna is the official language of Tigray region of Ethiopia and is the language most widely spoken in this region. It is also the national language of Eritrea which is the neighboring country of Ethiopia. Hence it is a medium of instruction of primary schools in the aforementioned areas and other Tigrigna speaking people [6, 12, 13].

Tigrigna's script has phonetic nature that has 249 characters, known as fidels or Abugida, orthographically. This orthographic representation of the language is organized into seven orders borrowed from Geez [13]. These seven orders are grouped by concatenating consonant and vowel. The following are the first two rows of seven ordered fidels.

Hä, hu, hi, ha, he, hə, ho / ሀ፡ ሁ፡ ሂ፡ ሃ፡ ሄ፡ ህ፡ ሆ

Lä, lu, li, la, le, lə, lo / ለ፡ ሉ፡ ሊ፡ ላ፡ ሌ፡ ል፡ ሎ

Table 1: Sample Seven Orders of Tigrinya Graphemes (Fidels)

	ä	U	i	a	e	(ə)	O
H	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
Xl	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ḥ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
M	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ

Text to speech systems have two main parts known as natural language processing and digital signal processing.

1.2 Natural Language Processing (NLP)

This is a module that involves text analysis, phonetic analysis and prosodic generation, and is responsible to make the text ready to be used by the synthesis module [2].

An instance of NLP module is Grapheme based synthesizer that is integrated in the Festvox voice building tool. It works for Unicode based orthographic languages using Unitrantransliteration tool [10]. It is more efficient for languages that have phonetic natured writing systems such as Spanish and Indic. Tigrinya is one of those languages that have direct correspondence between its letters and sounds of those letters.

1.3 Digital Signal Processing (DSP)

Digital signal processing is concerned with application of algorithms and models to generate synthetic speech from text manipulated by the NLP module [8, 12]. This is a backend part of speech synthesis also termed as speech synthesis method.

Text to Speech System Methods can be generally grouped into three main categories known as parametric speech synthesis, concatenative speech synthesis and statistical parametric speech synthesis according to their behavior of conversion of phone sequence to speech waveform [9].

Parametric synthesis systems (SPSS) parameters prediction coefficients are extracted from speech signal for each phone unit. Speech parameters are derived manually.

Concatenative synthesis method synthesizes based on the concatenation of prerecorded speech segments. This technique alleviates the drawback of parametric synthesis method which is manual derivation of parameters. However, it is limited to one speaker and one voice and usually requires more memory capacity than the other synthesizers [3, 12].

The third backend method, which is statistical parametric synthesis, is described as speech synthesis method that enables generating an average of some set of similarly sounding speech segments using parametric models [4]. This method differs from the traditional parametric method by using machine learning techniques to learn the parameters and any associated rules of co-articulation and prosody from speech data. It also differs from concatenative method since it does not use prerecorded speech segments. Instead it uses parameters that are extracted from speech signal [4]. Generally, SPSS is model based but the other two are not. SPSS has two distinct phases termed as training and synthesis phases.

One instance of SPSS is CLUSTERGEN Synthesis where its framework is integrated in Festival Speech Synthesis System [1]. It is a method for training models and using these models at synthesis time. The training requires well recorded utterances, and text transcriptions of what has been said.

2. Related Work

Sitaram *et al*. [10] developed Universal Grapheme-based Speech Synthesis. They tested it on 12 languages such as English and German from resourced languages, and Tamil and Thai from under resourced languages. The results were remarkable, especially for languages that have one to one mapping between their orthography and phoneme. They used CLUSTERGEN's Mel Cepstral Distortion (MCD) objective measure for their synthesized speeches. The lowest (that has good quality of synthesized speech) was 4.30 which is German while the highest was 6.71 which is Ojibwe.

Tesfay Yihdego [15] developed a prototype for Tigrigna Language using TTS System that is Diphone based concatenative speech synthesis. Diphones are used as the basic concatenation units to synthesize sample Tigrigna texts and also Time-Domain Pitch Synchronous Overlap and Add (TD-PSOLA) is used as a technique to generate the synthetic speech.

Unit selection based Text-to-Speech (TTS) system for Tigrinya using the Festival framework and practical applications of it was done by Agazi Kiflu [7]. The result was generating speech from the prepared corpus via selecting the needed units. Another work by Lemlem Hagos [5] developed Tigrigna Text to Speech Synthesizer. A prototype is developed using Festival Speech Synthesis Framework.

All the reviewed works are concatenative speech synthesis method based that is advantageous in high quality speech. But due to its consumption of enormous costs and time for constructing corpora and is not straightforward to synthesize diverse speakers' such as emotions, styles and the like, it becomes less favourable for under resourced languages and implementation in different application areas [11, 14]. In addition, they all transliterated Tigrinya letters to Latin alphabets and then to IPA instead of directly to IPA which adds extra process.

3. The Proposed Solution

The proposed solution of the gap shown in the previous attempts of synthesizing Tigrinya was aimed to the transliteration from Geez character to Latin Alphabet by making it directly to IPA and changing the backend to model based synthesizer which is CLUSTERGEN. The other solution to the gap was considering Tigrinya writing system as main base for synthesis instead of its phonological phonemes (32 phonemes). The proposed architectural view is shown in Figure 1 to indicate how the whole synthesis process looks like.

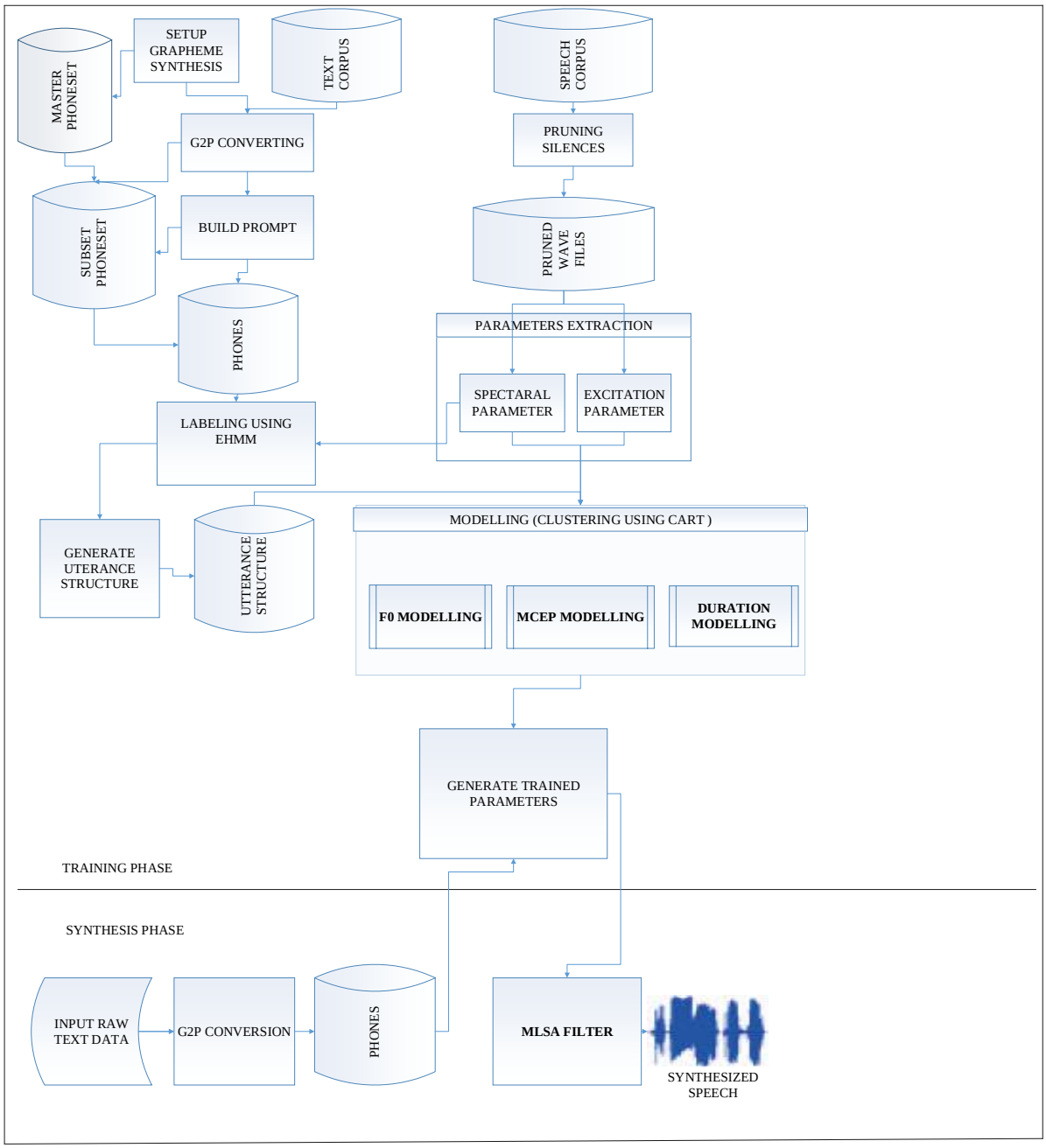

Figure 1: The Proposed Tigrinya General Speech Synthesis System Architecture

Eight hundred sentences that were previously prepared but recorded again in professional studio by native experienced female journalists was used as corpus.

Different software toolkits from recording to synthesis were used. Dell Laptop Core i7 that has 4 Gigabyte RAM and dual operating system, 32 bit Windows 10 and 32 bit Ubuntu 14.04 LTS was used in order to run this software. Wired microphone that had earphone was used for recording purpose while PRAAT was used as recording tool.

The list of software used, in which all are free, are: PRAAT, speech_tools-2.5.0-current.tar.gz, festival-2.5.0-current.tar.gz, festlex_CMU.tar.gz, festlex_POSLEX.tar.gz, festvox_kallpc16k.tar.gz, festvox-2.7.3-current.tar.gz and Speech Signal Processing Toolkit (SPTK-3.6.tar.gz). These software, except PRAAT, which was installed on Windows, were installed on Ubuntu.

4. Discussion

Preprocessing activities such as removing byte order and marker order from UTF-8 based Tigrinya

characters and preparing the corpus was done initially. The whole synthesis process was done using an incremental development process where first 200 utterances (sentences) were synthesized and in the next doubling of the first utterances were added, till it reached 800.

From the prepared text corpus phoneset was derived using the Grapheme based synthesizer frontend system. However, the phoneset was not full enough since it ignored 14 pharyngeal consonants. Tigrinya's 14 consonant characters which are pharyngeal voiceless fricatives (ሐ, ሑ, ሒ, ሓ, ሔ, ሕ) and pharyngeal voiced fricatives (ዐ, ዑ, ዒ, ዓ, ዕ, ዖ) were added to make the synthesis system efficient and effective. A phone '&' derived automatically also caused a problem, in the next steep, while labelling. This was due to the occurrence of this phone almost in every word of the language. This phone is changed to AMP and automatic labelling using Embed Hidden Markov labeler (EHMM) was done successfully. This was the beginning of training of Tigrinya speech synthesis. Spectral and excitation parameters were extracted using SPTK from the parallel speech corpus. These extracted parameters are used to develop three models named Spectral (F0) model, Excitation (MCEP) model and Duration model using Classification And Regression Tree (CART) via SPTK. Afterwards, these models are combined into one to use for synthesis phase. The text corpus was separated into training and testing, based on tenfold threshold. In the synthesis phase from the test dataset phones were derived automatically and these phones were aligned acoustic features from the combined trained model. Finally, synthesized speech was enabled using Mel Log Spectrum Approximation (MLSA) vocoder.

For the synthesized speech an objective measure, MCD, was generated automatically.

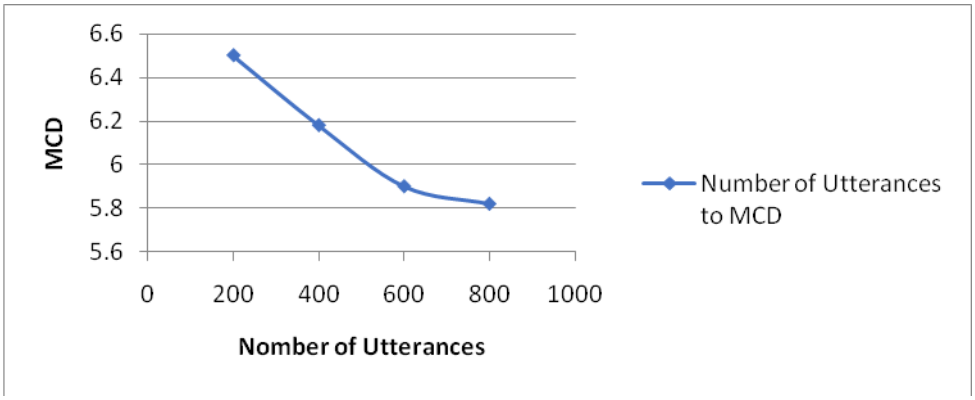

Figure 2: Experimental Results (for Objective Evaluation)

As shown in Figure 2, MCD of final 800 sentences was 5.82 which can be interpreted as the synthesized speech almost as the same as natural human speech sound in its naturalness and intelligibility.

Subjective evaluation using mean opinion score (MOS) that was done by 8 native speakers for randomly selected 10 sentences also resulted the same to MCD when interpreted. These evaluators were 4 males and 4 females aged between 18 and 64, which were diploma to master educational levels. Naturalness was evaluated as 4.8 while intelligibility was 4.3 out of 5.

5. Conclusion and Future Work

Lack of efficiency and effectiveness of synthesizing Tigrinya language was identified as a gap. Based on the identified gap, we developed a model-based speech synthesis prototype which is integrated into one comprehensive framework. The main reason for selecting this framework was the under resourced linguistic nature of the language. The Unicode encoded characters (orthography) of Tigrinya were considered as phonemes irrespective of the phonology of the language. The text corpus was existing. However, recording (speech corpus)

was done professionally in a recommended way for speech synthesis.

The whole end to end synthesis was done automatically using Grapheme based synthesis and CLUSTERGEN on Festival speech synthesis system. The synthesis methodology used on this study is an adoption of a well-accepted system, that had got recognition from other phonetic nature languages and for the sake of this study we made some additions and amendments.

Direct transcribing Tigrinya letters to IPA, that alleviated transliteration of the characters from Tigrinya letters to Latin that had been used in earlier works was tested and enabled; additions of 14 pharyngeal letters to the system derived Tigrinya phoneset was made; modification of one phone that frequently occurs in almost all words of Tigrinya that is used in the case of vowel insertion was changed from '&' to 'AMP'. A prototype that could possibly exploitable flite voice which could run on any Android device was developed.

Generally, an efficient, effective and model based speech synthesis prototype is developed (in Festival speech synthesis system) that has remarkable natural and intelligible speech. The test and evaluation results of the prototype have shown promising possibility of developing model-based Tigrinya speech synthesizer.

As future work, adding gemination handler module, using phonetically balanced and prosodically rich corpus and adding interpolation for a variety of speaking styles will be done for better result.

References

[1] Black, A. W., "CLUSTERGEN: A Statistical Parametric Synthesizer Using Trajectory Modeling", In Ninth International Conference on Spoken Language Processing, 2006.

[2] Black, A. W. and Lenzo, K. A., "Building Synthetic Voices", Language Technologies Institute, Carnegie Mellon University, 2014.

[3] Black, A. W. and Taylor, P. A., "Automatically Clustering Similar Units for Unit Selection in Speech Synthesis", 1997.

[4] Black, A. W., Zen, H., and Tokuda, K., "Statistical Parametric Speech Synthesis", In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, 2007.

[5] Lemlem Hagos, "Developing Tigrigna Text to Speech Synthesizer", Unpublished PhD Dissertation, Addis Ababa University, 2015.

[6] Kievit, D. and Kievit, S., "Differential Object Marking in Tigrinya", Journal of African Languages and Linguistics, 2009.

[7] Agazi Kiflu and Tibebe Beshah, "Unit Selection Based Text-to-Speech Synthesizer for Tigrinya Language", HiLCoE Journal of Computer Science and Technology,Vol 1. No. 1, 2013.

[8] Prahallad, K., "Automatic Building of Synthetic Voices from Audio Books, Unpublished PhD Dissertation, Carnegie Mellon University, 2010.

[9] Shinoda, K. and Watanabe, T., "MDL-based Context-dependent Subword Modeling for Speech Recognition, 2001.

[10] Sitaram, S., Parlikar, A., Anumanchipalli, G. K., and Black, A. W., Universal Grapheme-based Speech Synthesis", In the Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[11] Taylor, P., "Text-to-Speech Synthesis", Cambridge University Press, 2009.

[12] Tesfay Tewolde, "A Modern Grammar of Tigrinya", Rome, Italy, 2002.

[13] Tigrigna Ethnologue, www.ethnologue/ language/tir, Last Accessed on 27 November, 2016.

[14] Yamagishi, J., "An Introduction to HMM-Based Speech Synthesis", Technical Report, 2006.

[15] Tesfay Yihdego, "Diphone Based Text-to-Speech Synthesis System for Tigrigna", Unpublished Masters Thesis, Addis Ababa University, 2004.

[16] Zen, H., Tokuda, K., and Black, A. W., "Statistical Parametric Speech Synthesis", Speech Communication, 51(11), pp. 1039-1064, 2009.