महस्विनी – विद्वत्समीक्षिता सन्दर्भिता च शोधपत्रिका (ISSN: 2231-0452)
कुसुमम् : जनवरी – डिसेम्बर २०२० सम्पुटम् : संयुक्तम् (१,२)

Implementing and evaluation of Computer based speechification for Sanskrit Text Corpus

Dr. B. Chandrasekharam*

Abstract: National Sanskrit University has created a rich Sanskrit text corpus with search tools and made it available at http://sansknet.ac.in. To enhance and extend the functionality of the website it is thought of integrating speechification facility with the website. Hence a study was undertaken to identify evaluate existing python based open source modules. This paper discusses implementation of Speechification/TTS system for Sanskrit Text Corpus using python modules developed by various institutions/individuals under different projects. This paper also evaluates tangibility, nativity, roboticness of speechified output produced by employing various modules developed using Likert Scale and made available at http://pypi.org. Python Package Index (PyPI) is a repository of software for the Python programming language[1]. This paper selectively considered to employ and evaluate python packages: sanskrit_tts, nltk, inltkgTTS, pyTTS, after thoroughtly surveying the available NLP, Text to Speech related python packages.

Keywords: TTS, Speechification, Sanskrit Text Corpus, NLTK, inltk, sanskrit_tts, pyTTSLikert Scale

1. Introduction

Rashtirya Sanskrit Vidyapeetha, Tirupatin now known as National Sanskrit University created a rich digitized rare Sanskrit text corpus in

* Associate Professor, Dept. of Computer Science, National Sanskrit University, Tirupati

[1] https://pypi.org/, a repository of software for the Python programming language.

ASCII format and made it available on internet at web link http://sansknet.ac.in/p[2]. This web resource maintained by National Sanskrit University provides access to more than six lakh electronic pages, which were created by data entry operators using traditional data entry methods on GIST terminals, covering many areas of Sanskrit Literature. An electronic page is considered to have 2000 key strokes. With the shift to online medium during the pandemic, and the gradual build-up to the digital world of the 21st century, the process of digitalization has transformed countries.Enabling institutions and transforming processes to the digital medium constitutes digitalization. Sanskrit University authorities felt a strong need for providing Sanskrit Digital Learning/Text Resources to its stake holders as part of extending reading material for curriculum needs of the students. The pandemic situation forced Teaching and learning activities to become Screen based. Being constrained by the situation experienced by Sanskrit learners during the pandemic, the author was motivated to work for speechifying the existing digitized Sanskrit text corpus[1].

Sanskrit Learner studying in traditional and online classroom settings must primarily rely on their listening and reading ability while attending live classes, specifically if the lecture is not available on video. Virtual courses enable learners to listen to the lecture online. Recording/ speechifying and providing digital text learning material enable learner to read and listen to the speechified Sanskrit text to better comprehend and revise the topic at their own speed. In either setting, educational institutions can collaborate with transcription services to help organize and present the information to students to promote better learning.

[1] http://sansknet.ac.in Unicoded Sanskrit text corpus developed at National Sanskrit University

Any traditional Text to Speech/ Speechification system consists of the 1) Natural Language Processing, and 2) Digital Signal Processing Module. Two primary objectives of digitization are to provide access and for purposes of preservation. For access, a priority is extending reach geographically. For preservation, fragile records can be stored for years without much maintenance in digital storage. Fig 1 shows the three Ds-digitization, digitalization, and digital transformation and how they are interlinked.

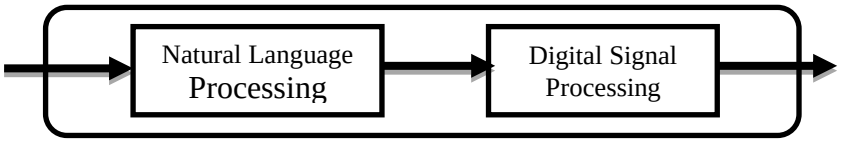

Fig.1.Simple Block diagramSpeechification

2. Python based Text to Speech Processing packages

gTTS:There are several APIs available to convert text to speech in Python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS (Google Text-to-Speech), a Python library

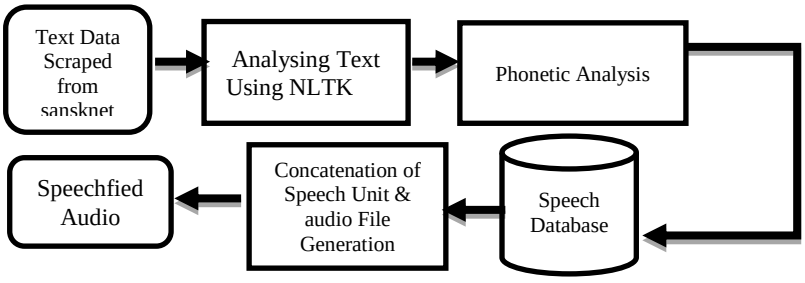

Figure 2 Various stages in Speechfication Process

and CLI tool to interface with Google Translate's text-to-speech API is a very easy to use tool which converts the text entered into audio which can be saved as a mp3 file. The gTTS python package available at

pypi.org website suppors several languages including English, Hindi, Telugu, Tamil, Kanada, French, German and many more. The package creates audio in two available audio speeds fast or slow. It does not directly support Sanskrit language.

Sanskrit_tts: A simple python library for converting Sanskrit text to speech. Currently, the only supported method is to use Google Cloud text to speech with a work around. A Google Cloud account with text-to-speech API enabled is required. It may be mentioned here that the TTS API will incur a cost if the free quota is exceeded (details). The library uses a wavenet voice by default, but this can be changed via the voice parameter to synthesize_text. This is an extended implementation of gTTS python Package.

Features: 1. Customizable speech-specific sentence tokenizer that allows for unlimited lengths of text to be read, all while keeping proper intonation, abbreviations, decimals and more; 2. Customizable text pre-processors which can, for example, provide pronunciation corrections;

```
pip install sanskrit_tts
import os
from IPython.display import Audio
from sanskrit_tts import gcloud_tts as tts
# Setup credentials
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = './credentials.json'
text =
"""कालिदासस्यजीवनवृत्तिविषयेअनेकाःलोकविश्रुतयःअनेकेवादाःचसन्ति।केचित्एनंविक्रमादित्यस्यसभायांकविःइतिमन्यन्ते।केचित्गुप्तकालीननरेशाणाम्आश्रयंप्राप्तवानितिकथयन्ति।धारानगरेभोजराजस्यसभायांकविरत्नपदभूषितःअभूत्इतिकथाकोविदाःकथयन्ति।जनश्रुत्यनुसारंबाल्यकालेसःअतीवमूर्खःआसीत्।विद्याधरयासहत
स्यविवाहःअभवत्। """
audio = tts.synthesize_text(text)
Audio(list(audio.get_array_of_samples()), rate=audio.frame_rate, embed=True)
```

Figure 3 Sanskrit TTS installation instructions

Pyttx3package: Python Text to Speech:pyttx module: this is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline. The pyttx3 library is compatible with both Python 2

and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3.Engine instance. During construction, the engine initializes a pyttsx3.driver. Driver Proxy object responsible for loading a speech engine driver implementation from the pyttsx3.drivers module. After construction, an application uses the engine object to register and unregister event callbacks; produce and stop speech; get and set speech engine properties; and start and stop event loops.

```
import pyttsx3
engine = pyttsx3.init() # object creation
""" RATE"""
rate = engine.getProperty('rate')   # getting details of current speaking rate
engine.setProperty('rate', 125)     # setting up new voice rate
"""VOLUME"""
volume = engine.getProperty('volume')   #getting to know current volume level (min=0 and max=1)
engine.setProperty('volume',1.0)    # setting up volume level  between 0 and 1
"""VOICE"""
voices = engine.getProperty('voices')       #getting details of current voice
engine.setProperty('voice', voices[1].id)   #changing index, changes voices. 1 for female
engine.say("hari om")
engine.say('My current speaking rate is ' + str(rate))
engine.runAndWait()
engine.stop()
"""Saving Voice to a file"""
engine.save_to_file('hari om', 'test.mp3')
engine.runAndWait()
```

Figure 4 Py-TTX3 Installation

To install and test pyttx3 pacakge the following statements are used:

5. Issues identified while integrating and implementing python based modules:

The py_ttx3 package does not support Devanagari script directly and hence wx_conv python package was used to transliterate to roman script for inputting the py_ttx3 module.

Google Text-to-Speech does not support Sanskrit yet. As a work around, this library uses other languages for speech to text conversion. Kannada is used by default for this workaround. Any other language/voice supported by Google TTS API can be used by changing

the voice parameter to synthesize_text, and the results will vary. While experimenting the voice parameter was set to different values to identify best work around for Sanskrit language.

While implementing the python speech modules, the following aspects/features of python speech modules under implementation should be acceptable to Sanskrit Scholar community inorder to integrate with Sanskrit text corpus.

1) tangibility of speechified output obtained from implementation
2) nativity of speechified output obtained from implementation
3) Naturality/No roboticness of speechified output obtained from implementation.

6. Evaluation of output of the implemented Modules:

Speech quality is a complex psychoacoustic outcome of the human perception process. As such, it is necessarily subjective, and can be assessed through listening test involving human test subjects that listen to a speech sample and assign a rating to it[1]. Users of new speech processing applications are often unaware of the underlying technology. Their main criterion for assessing these application is based on overall speech quality. Therefore, this paper considers three subjective audio metric tests. As a part of these tests, the private web link is shared to scholarly group of Sanskrit listeners who are asked to rate the quality of output of the respective modules based on overall perception. In this paper Likert Psycho metric scale is employed to evaluate the output of implemented speechified module.

Likert scale: Psychometric scale known as Likert scale is used to evaluate the Speechification implementation. Likert Scale is the most

[1] Grncharov, W.B.Kleijn, Speech Quality Assessment, Benesty, Sondhi, Huang, Handbook of Speech Processing, Springer-Verlag Berlin Heidelberg-2008

widely used approach to scaling responses in survey research when compared to other types of rating scales[1]. As the quality of synthesized speech depends subjective perception of listener, Likert scale is chosen to evaluate the implementation.

To carryout evaluation, a five minute synthetic speech output generated by pyTTX3 and Sanskrit_TTs modules were presented to 100 Sanskrit Scholars for scaling the speech quality with respect to tangibility, nativity, roboticness on Likert Scale.The following table relates to subjective evaluation made by Sanskrit listeners

	Sanskirts TTS		PY TTS/py_ttx3		PYTTS/ py_ttx3	Sanskirts TTS
	Tang-ibility	Nativity	Tang-ibility	Nativity	Robotic ness	Robotic ness
Satisfied/ Agreed	88	83	65	45	82	87
dissatisfied/ disagree	1	7	16	35	2	1
Netural	2	2	5	2	5	3
Stongly Satisfied	7	2	0	0	8	8
Strongly dissatisfied	2	6	14	18	3	1

Figure 5 Likert Scale rating -Subjective Tests for PyTTS and Sanskrit_TTS

During the implementation phase itself several combinations with each module was implemented with several python virtual environments. Each virtual environment was implemented by setting different parametres like speed, voice, gender. Experiment was repeated by chaging NLP packages NLTK and inltk. The inltk Natural Language

[1] https://en.wikipedia.org/wiki/Likert_scale

Processing module resulted in lot of errors while integrating and hence was not implemented here.

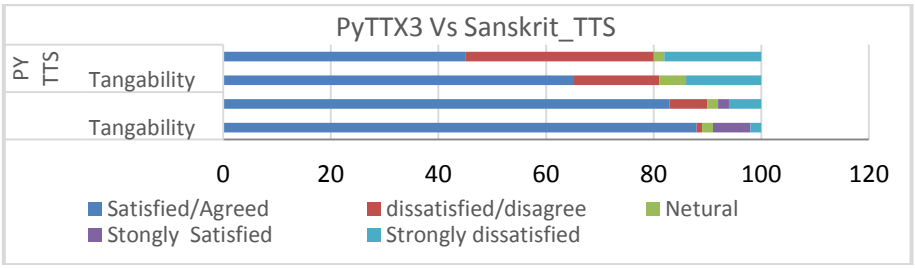

Figure 6 Likert scale- Nativity & Tangibility Evaluation

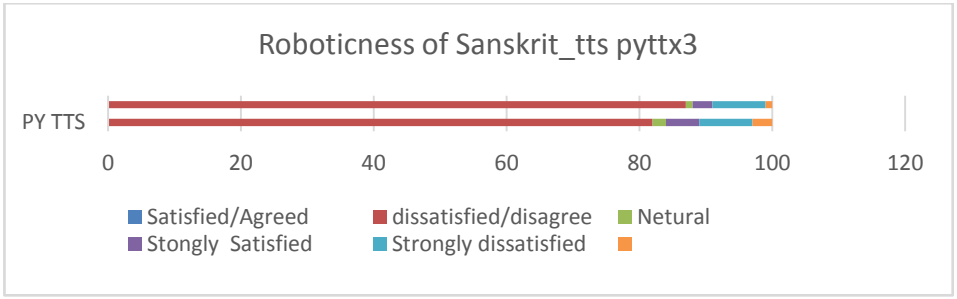

Figure 7 Likert scale- Roboticness

7. Conclusion: An experiment was conducted by implementing Sanskrit_tts and pyttx3 modules for Sanskrit text speechification and to identify a best tts module from python packages repository. Both the packages were subjectively evaluated using Likert Psycho metrics as shown in above. From the evaluation Sanskrit_TTS package performed better for integration with Sanskrit Text Corpus for computer based speechification.

❑❑❑