INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
*(Volume 5, Issue 6)*
Available online at: www.ijariit.com

# Survey on text-to-speech Kannada using Neural Networks

*Pawan S. Nadig*
*pawan.nadig@gmail.com*
*Vidyavardhaka College of Engineering, Mysore, Karnataka*

*Pooja G.*
*pooja.gl598@gmail.com*
*Vidyavardhaka College of Engineering, Mysore, Karnataka*

*Kavya D.*
*dkavyaachar@gmail.com*
*Vidyavardhaka College of Engineering, Mysore, Karnataka*

*R. Chaithra*
*rchaithra086@gmail.com*
*Vidyavardhaka College of Engineering, Mysore, Karnataka*

*Radhika A. D.*
*radhika.ad@vvce.ac.in*
*Vidyavardhaka College of Engineering, Mysore, Karnataka*

## ABSTRACT

***In this paper, we have explained the different approaches of text-to-speech conversion for Kannada language using Neural Networks. Text-to-speech conversion (TTS) has a wide variety of applications like it serves as an effective aid to the visually impaired by offering a computer-generated spoken voice that would "read" text to the user. TTS can also be used as an interactive educational appliance, as an assistant, etc. The application field of synthetic speech is expanding fast while the quality of the TTS system is also increasing steadily. Text-To-Speech conversion using Neural Networks is found to be more accurate when compared to many other methods like Concatenation, Digital Signal Processing, Articulatory Synthesis, etc. The different approaches of neural networks that we have mentioned in this paper explain its own methodologies, its applications and also its pros and cons.***

***Keywords***— *Neural Networks, Kannada, TTS, Approaches*

## 1. INTRODUCTION

In this paper, the approaches explained are based on neural networks. Neural Networks is a circuit of neurons, composed of artificial neurons or nodes. These artificial networks are used for predictive modeling, adaptive control, and applications where they can be trained via dataset. In this paper we have explained Neural Networks approach using different types of Neural Networks like DNN, RNN [1], and CNN, etc. Deep Learning models are very good at learning features from data, i.e., if TTS wants to solve: we have to input a text, and the output should be audio, in this case, the audio can be represented as a waveform. TTS conversion using neural networks approach is more accurate when compared to all other approaches. Since Neural Networks are more scalable and precise proper computations of the results can be achieved.

## 2. EXISTING SYSTEM

Most of the existing text-to-speech conversion systems are mainly built for the English language and very few TTS systems are built for regional languages like Assamese, Telugu, Tamil, Kannada, and Hindi, etc. The current TTS systems for Kannada language are based on Concatenation approach. The processing of Kannada text in most of the existing systems is not accurate due to lack of proper datasets and the methodologies used. The result obtained in these systems is improper because of the following problems: correct prosody and pronunciation analysis from the written text is a major problem, written text contains no explicit emotions, digits and numerals cause other types of problem when included in the text i.e., TTS systems fails to expand the numbers or sometimes it may skip, adding special symbols like '-' may not predict the proper results since it may consider it as a range or a mathematical functional operator etc. These problems have made the existing TTS systems less efficient.

## 3. PROPOSED SYSTEM

To overcome the problems of the existing system we are trying to implement a more accurate and efficient Text-To-Speech system for Kannada language using Neural Networks approach.

## 4. RELATED WORK

**"Duration modelling for the Telugu Language with Recurrent Neural Network" by V. S. Ramesh Bonda, P. N. Girija [1]:** This paper makes use of novel syllable duration modeling approach. First the input text is normalized by expanding abbreviations, acronyms, numbers and all non-standard words. It has been shown that RNNs use short term memory. In RNN the connections between units form a directed cycle which allows it to exhibit dynamic temporal behavior. RNN is more superior in learning many behaviours / sequence processing tasks/algorithms/ programs. Compared to traditional machine learning methods. A feed-forward neural network is used to predict duration for Telugu.

"**Speech Synthesizer for Tamil using Neural Networks" by K. Lekshmi1, K. Ruba Soundar [2]:** In this method, speech is produced by selecting and concatenating appropriate speech unit

 

from speech database. A speech database consists of speech units of different sizes such as phones, diphones, syllables, words or sentences. Speech synthesizer is used here for Tamil which reads the text in the Tamil language. A TTS engine consists of two parts: a) synthesizer b) parser. Synthesizer: Speech synthesizer gets the text document, it parses that document to extract the text and it sends the text to the synthesizer and plays the synthesized speech. Synthesizer is in the form of a plugin. Plug in takes text document as input and intelligible Tamil speech as output. This research is done mainly to improve the human interface to the computer.

**"Merlin: An Open Source Neural Network Speech Synthesis System" by Zhizheng Wu, Oliver Watts, and Simon King [3]:** This paper introduces the Merlin speech synthesis toolkit for neural network-based speech synthesis. The system takes linguistic features as input and employs neural networks to predict acoustic features, which are then passed to a vocoder to produce the speech waveform. The toolkit is Open Source, written in Python, and is extensible. Freely-available tool-kits are available for two of the most widely used methods: waveform concatenation and HMM-based statistical parametric speech synthesis, or simply SPSS.

**"A study of speaker adaptation for DNN-based speech synthesis" by Zhizheng Wu, Oliver Watts, and Simon King [4]:** In this paper, they are conducting an experimental analysis of speaker adaptation for DNN-based speech synthesis at different levels. In DNN synthesis, when training a speaker-independent DNN model- an average voice model (AVM) – linguistic features are added with i-vector as an additional to capture speaker identity. At the adaptation phase, the target speaker's i-vector is first estimated by using the adaptation data and the total variability T through $s \approx m + Ti$, $i \sim N(0,I)$ this equation, and then the i-vector is appended with linguistic features as input to generate the target speaker's voice. Learning Hidden Unit Contribution (LHUC) is a method that linearly re-combines hidden units in a speaker- or environment-dependent manner using small amounts of unsupervised adaptation data. They extend LHUC to a speaker adaptive training (SAT) framework that leads to a more adaptable DNN acoustic model. At the output level, we perform a feature space transformation to modify the output of a DNN as $y \approx F(y0)$, where y0 is the output feature of a DNN, y is the reference target vocoder parameter, and F (.) is a transformation function. They focus on improving the naturalness of speech synthesis.

**"Deep Voice 3: 2000-Speaker Neural Text-To-Speech" by Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arık, Ajay Kannan, Sharan Narang, Jonathan Raiman, John Miller [5]:** This paper proposes a fully-convolutional character-to-spectrogram architecture, which enables fully parallel computation and trains an order of magnitude faster than analogous architectures using recurrent cells. It shows that the architecture trains quickly and scales to the LibriSpeech ASR dataset which consists of 820 hours of audio data from 2484 speakers. This also demonstrates that they can generate monotonic attention behavior, avoiding error modes commonly affecting sequence-to-sequence models. They compare the quality of several waveform synthesis methods, including WORLD, Griffin-Lim, and WaveNet. It describes the implementation of an inference kernel for Deep Voice 3, which can serve up to ten million queries per day on one single-GPU server.

**"Tacotron: Towards End-To-End Speech Synthesis" by YuxuanWang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weissy , Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengioy, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous [6]:** The backbone of Tacotron is a seq2seq model which includes an encoder, an attention-based decoder, and a post-processing net. At a high-level, our model takes characters as input and produces spectrogram frames, which are then converted to waveforms. The convolution outputs are fed into a multi-layer highway network to extract high-level features. Finally, we stack a bidirectional GRU RNN on top to extract sequential features from both forward and backward context. Encoder -The goal of the encoder is to extract robust sequential representations of text. The input to the encoder is a character sequence, where each character is represented as a one-hot vector and embedded into a continuous vector. We then apply a set of non-linear transformations, collectively called a "pre-net", to each embedding, which helps convergence and improves generalization. The Griffin-Lim algorithm is used to synthesize waveform from the predicted spectrogram.

## 5. CONCLUSION

By making research on all the techniques, we found Neural Networks approach is more accurate to obtain the proper results with fewer errors. The research is further continued to select the appropriate methodology to design the Text-to-speech conversion system for Kannada language. This paper explains the functionalities, applications, pros, and cons of different types and techniques of Neural Networks which is helpful to select the best approach to work on TTS system.

## 6. REFERENCES


[1] V. S. Ramesh Bonda, P. N. Girija, "Duration Modeling For the Telugu Language with Recurrent Neural Network", February 2015.

[2] K. Lekshmi, K. Ruba Soundar, "Speech Synthesizer for Tamil using Neural Networks", October 2011.

[3] Zhizheng Wu, Oliver Watts, Simon King, "Merlin: An Open Source Neural Network Speech Synthesis System", 13-15 September 2016.

[4] Zhizheng Wu, Pawel Swietojanski, Christophe Veaux, Steve Renals, Simon King, "A study of speaker adaptation for DNN-based speech synthesis", 14 September 2015.

[5] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arık, Ajay Kannan, Sharan Narang, Jonathan Raiman, John Miller, "DEEP VOICE 3: 2000-SPEAKER NEURAL TEXT-TO-SPEECH", 2018.

[6] YuxuanWang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weissy , Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengioy, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous, "TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS", 6 April 2017.

[7] Anand H. Unnibhavi, D. S. Jangamshetti, "Development of Kannada Speech Corpus for Continuous Speech Recognition", June 2018.

[8] "Problems in Speech Synthesis", http://research.spa.aalto.fi

[9] "Speech Synthesis", http://en.wikipedia.org

[10] [10]"Speech Synthesis Techniques using DNN" https://medium.com


 