==========
Language in India www.languageinindia.com ISSN 1930-2940 Vol. 24:4 April 2024
==========

Malayalam Text-to-Speech Conversion: An Assistive Tool for Visually Impaired People

Muhammed Shafi M
Ph.D., Research Scholar, Department of Linguistics
University of Kerala, Thiruvananthapuram -695581
Email: shafim.lin@keralauniversity.ac.in

Prof. (Dr.) S.A. Shanavas
Professor and Head, Department of Linguistics
University of Kerala, Thiruvananthapuram -695581
Email: sashanavas@keralauniversity.ac.in

==========

Abstract

The discourse on Malayalam Text-to-Speech (TTS) Conversion emphasizes its crucial role as an assistive tool for the visually impaired, addressing the challenges they face in accessing printed and digital content. By enabling synthesized speech, Malayalam TTS technology enhances accessibility and inclusivity, allowing visually impaired individuals to engage with digital content independently. The flexibility of Malayalam TTS, including language preference and pacing options, facilitates efficient information consumption for users. Its applications in education and professional environments illustrate its role in levelling the playing field for visually impaired individuals. However, challenges such as adoption and usability persist, necessitating improvements in TTS quality and compatibility, alongside efforts to promote accessibility standards. Ultimately, Malayalam TTS serves as a means of empowerment, providing equitable access to information and fostering personal and professional development for the visually impaired community.

Keywords: Text-to-Speech Conversion, Visually Impaired People, Assistive tool, Malayalam.

Introduction

Nowadays, digital communication and the exchange of information play an ever-growing role, ensuring accessibility for everyone, including individuals with visual impairments, is absolutely essential. For speakers of Malayalam, a language with a rich cultural heritage, accessing digital content can pose significant challenges. Malayalam is a Dravidian language spoken predominantly in the Indian state of Kerala and some parts of

neighbouring states. However, the development of Malayalam text-to-speech (TTS) conversion technology offers a transformative solution, serving as a powerful assistive tool for visually impaired individuals. In this article, we explore the role of Malayalam TTS conversion in enhancing accessibility and inclusivity for the visually impaired community. This technology offers customizable speech settings, improving comprehension and efficiency in information consumption. It also supports multi-modal interactions, benefiting education and workplace applications, and promotes inclusivity by levelling the playing field for visually impaired individuals. However, challenges remain, such as enhancing TTS quality and compatibility with digital platforms, necessitating efforts to raise awareness and promote accessibility standards for broader adoption.

Review of Literature

Text-to-speech (TTS) systems play a crucial role in aiding visually impaired individuals by converting text from images into audio format for easier comprehension [1] [2]. These systems utilize technologies like Natural Language Processing (NLP) for accurate text-to-speech conversion [3] [4]. The aim is to provide a cost-effective and easily accessible solution for the visually impaired to read and understand text from various sources like newspapers or posters [5]. By employing machine learning algorithms and OCR tools, these TTS systems help in extracting text from images, processing it, and converting it into speech, thereby enhancing the daily lives of visually impaired individuals by enabling efficient text reading through audio output.

Arun Gopi et al. discusses the shift towards concatenative synthesis in text-to-speech (TTS) development, highlighting its advantages over parametric synthesis for higher quality output. It introduces the Epoch Synchronous Non-Overlap and Add (ESNOLA) technique for Malayalam TTS on the Android platform, utilizing diphone-like segments as basic units for concatenation from a database of 1500 partnemes. The implementation covers database generation, Android platform modifications, database access, and Malayalam character display. Additionally, the paper presents a Newsreader application design. The TTS system achieves a Mean Opinion Score (MOS) of 3.2 in perceptual tests, indicating satisfactory user perception[6]. For text to speech conversion the data provided for each language in IndicTTS is insufficient to effectively train more advanced neural-network-based TTS systems. This limitation was highlighted by Srivastava et al. in 2020 when they introduced IndicSpeech, a more extensive corpus specifically aimed at training neural TTS systems for three Indian languages[19].

Interestingly, Srivastava et al. observed differences in the performance of TTS models trained on different language corpora within IndicSpeech. Specifically, they found that the mean opinion score (MOS) obtained for the TTS model trained on the Malayalam corpus was lower compared to those trained on Hindi and Bengali corpora[20]. They attributed this discrepancy to the inherent characteristics of Malayalam, such as the morpho-phonemic changes that occur during word formation, which can pose challenges for TTS synthesis.

Architecture of Malayalam Speech Synthesiser

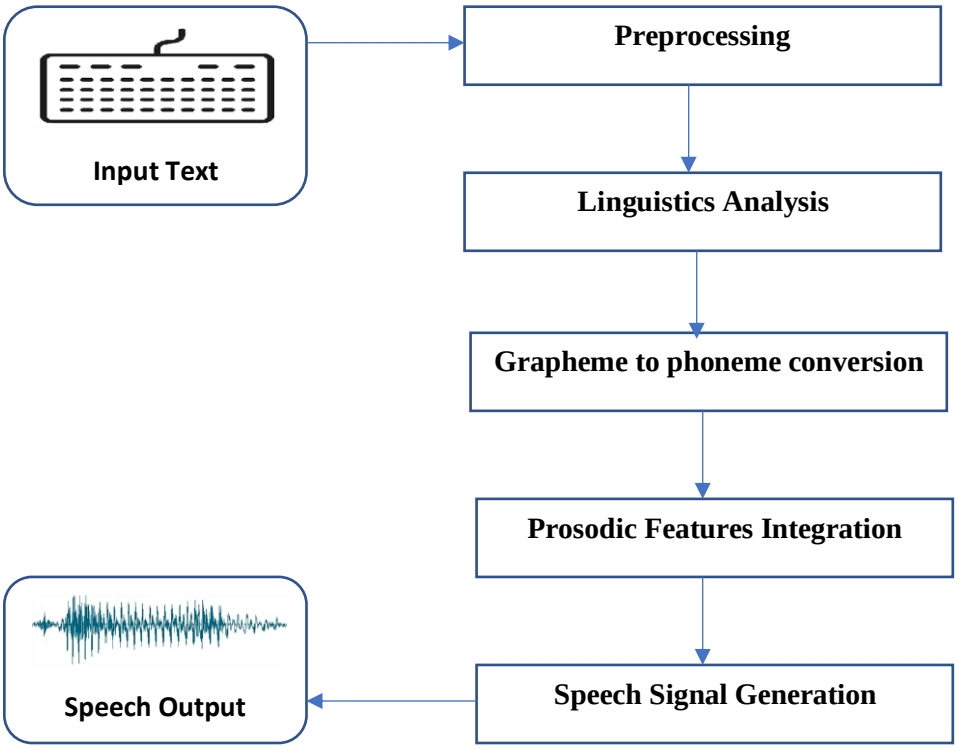

Figure 1 Architecture of Proposed System

The input console of the speech synthesizer system allows users, including those with visual impairments, to enter text using standard keyboard input or alternative methods such as speech recognition or braille keyboards. This console, whether part of the synthesizer application or integrated into other commonly used software like word processors or email clients, facilitates seamless text entry for visually impaired individuals. The system ensures accessibility by accommodating diverse text entry preferences and needs.

Text-to-speech (TTS) synthesis encompasses several pre-processing steps crucial for producing natural and comprehensible speech. These steps include text normalization, ensuring consistent pronunciation; tokenization, breaking text into smaller units for analysis; and part-of-speech tagging [14][15][16], assigning grammatical tags to aid in pronunciation and prosody determination. Linguistic analysis is integral in TTS synthesis, involving the examination of syntactic and semantic structures to grasp the text's meaning and context. This process entails parsing sentences, identifying grammatical dependencies, and resolving any ambiguities present. Another vital step is text-to-phoneme conversion, where words are mapped to their corresponding phoneme sequences, the fundamental units of speech sound. This conversion facilitates accurate pronunciation during synthesis by leveraging the phonetic content of the input text. These processes collectively ensure the production of natural and intelligible speech output in TTS systems.

Speech signal generation in a Text-to-Speech (TTS) system begins by converting text input into audible speech signals. This process entails concatenating selected phonemes to form larger speech units like diphones or triphones, ensuring natural transitions between speech sounds. Subsequently, signal processing techniques are applied to adjust parameters such as pitch, duration, and spectral characteristics, enhancing the naturalness of speech and minimizing distortion [11]. These processed speech units are then synthesized into a final speech waveform, representing the acoustic signal to be played back to the user. Finally, the synthesized speech waveform is converted into an audio format, such as PCM, and delivered through speakers or headphones, enabling the user to perceive the synthesized speech. Overall, this process involves converting text into processed speech units and synthesizing them to create natural-sounding speech for the user.

A Text-to-Speech (TTS) system comprises several modules to convert text into spoken speech seamlessly. It initiates with text processing, handling punctuation, formatting, and special characters, followed by passing the processed text to the speech synthesizer module. This module employs diverse techniques, such as generating phonetic representations, selecting speech units, and applying prosody, to produce natural-sounding speech [12]. Key components include the Text Analyzer Module for linguistic analysis, the Natural Language Processing (NLP) Unit for understanding text meaning and structure, the Synthesizer Module for generating speech based on phonetic representations, and the Partneme Database containing speech sound units for synthesis. These modules collaborate seamlessly to transform text input into synthesized speech, preserving naturalness and quality throughout the process.

Prosodic Features Integration

In linguistics, prosodic features refer to aspects of speech such as intonation, stress, rhythm, and tempo. These features play a crucial role in conveying meaning, emotion, and emphasis in spoken language. Adding prosodic features to a text involves annotating or tagging the text with information about these features. This annotation can be done manually or through automated processes using computational linguistics techniques [7]. Various methods exist to incorporate prosodic features into text. Speech processing algorithms are utilized to analyze audio recordings, extracting elements like pitch, intensity, duration, and rhythm. Text analysis focuses on linguistic features within the text itself, such as punctuation, word choice, and sentence structure, to infer prosodic patterns. Machine learning models are trained on annotated datasets to predict prosodic features directly from textual input. Natural language processing techniques delve into syntactic and semantic cues within the text that correlate with prosodic features [8]. Additionally, rule-based systems are developed to encode linguistic rules governing prosody, enabling the annotation of text accordingly. These approaches collectively facilitate the integration of prosodic information into text for various applications, ranging from speech synthesis to sentiment analysis.

Text	IPA	Tagset
കൊല്ലം	k	'plosive', 'voiceless', 'unaspirated', 'velar'
	o	'v_sign'
	l	'lateral', 'alveolar', 'virama'
	l	'lateral', 'alveolar'
	a	'inherentvowel'
	m	'anuswara'

Table 1: Represent Syllble classes in Malayalam Language

prosodic features play a significant role in conveying emotions in speech. By incorporating prosodic features such as intonation, stress, rhythm, and tempo into text, it becomes possible to add emotional nuance to written language. For example, variations in pitch and rhythm can convey excitement, while changes in tempo and stress may indicate tension or urgency. Additionally, prosodic cues can help convey subtler emotions such as sarcasm, empathy, or uncertainty. Integrating prosodic features associated with different emotional states enriches the expressiveness of text and enhances its ability to evoke appropriate emotional responses from readers or listeners. This integration is particularly important in applications such as dialogue systems, virtual assistants, and text-to-speech synthesis, where conveying emotions accurately can significantly improve user experience and communication effectiveness.

Speech Synthesiser

The synthesizer module in the Text-to-Speech (TTS) system identifies segments for concatenation based on phonetic strings representing actual pronunciations [17]. Token generation rules are applied to generate tokens used for identifying partnemes in concatenation [18]. These rules, which vary by language, guide token generation from preceding and succeeding phones. Tokens correspond to indexing of segmented partneme voice signals in the speech database header [9][10]. An offset calculation method is employed to determine byte information for segment retrieval, enhancing database search efficiency. Spectral smoothing is performed at concatenation points to reduce spectral disturbances, achieved through proper windowing of the output signal. The window for spectral smoothing is defined mathematically to ensure minimal distortion at concatenation points, facilitating smooth speech synthesis.

The token generation rules dictate the generation of tokens used for identifying partnemes in the concatenation process. These rules, which are language-specific, define how tokens are generated based on the configuration of preceding and succeeding phones. The provided token generation rules specify different patterns and configurations:

CVCV .. C + CV + V + VC + C + V + Vout (1)

VCV .. Vin + V + VC + C + CV + V + Vout (2)

CVYV .. C + CV + V + VY + YV + Vout (3)

CVV .. C + CV + VV + Vout (4)

C1C2V .. C1 + C2 + C2Vin + V + Vout (5)

1. For the pattern "CVCV," the token is generated by concatenating the preceding consonant (C), followed by a consonant-vowel pair (CV), another vowel (V), followed by a vowel-consonant pair (VC), another consonant (C), and finally, another vowel (V) as the output (Vout).
2. For the pattern "VCV," the token starts with a fade-in vowel (Vin), followed by a vowel (V), then a vowel-consonant pair (VC), a consonant (C), another consonant-vowel pair (CV), another vowel (V), and finally, the output (Vout).
3. For the pattern "CVYV," the token comprises a consonant (C), followed by a consonant-vowel pair (CV), a vowel (V), a vowel glide (VY), another glide-vowel pair (YV), and finally, the output (Vout).
4. For the pattern "CVV," the token consists of a consonant (C), followed by a consonant-vowel pair (CV), a double vowel (VV), and finally, the output (Vout).
5. For the pattern "C1C2V," the token is formed by concatenating two consonants (C1 and C2)

The offset calculation for tokens corresponding to vowels and consonants follows the given formulas:

1. For vowel tokens:

$$offset_{vowel}=(offset_{vi}-offset_{n})\times(S+I+V)\times B$$

$$offset_{vowel}=(offset_{vi}-offset_{n})\times(S+I+V)\times B$$

2. For consonant tokens:

$$offset_{consonant}=(offset_{ci}-offset_{n})\times(S+I+C)\times B$$

$$offset_{consonant}=(offset_{ci}-offset_{n})\times(S+I+C)\times B$$

Where:

- $offset_{vi}$offset$_{vi}$ and $offset_{ci}$offset$_{ci}$ represent the byte positions corresponding to the i^{th} vowel and consonant in the database, respectively.
- $offset_{no}$ffset$_{n}$ is the starting byte position of data.
- SS, II, and VV (or CC) represent the starting byte of data, vowel (or consonant), and consonant (or vowel) in the speech database, respectively.
- B denotes the byte size for each data.

These formulas are used to calculate the offset, which provides byte information for locating the samples corresponding to the partnemes in the speech database.

Figure 2 Wave form of the word 'Kollam'

Result and Discussions

The offset calculation method outlined by the provided formulas significantly enhances the efficiency and accuracy of speech signal generation in Text-to-Speech (TTS) systems. By precisely determining byte positions for accessing samples corresponding to vowels and consonants in the speech database, this method streamlines the synthesis process, reducing search time and optimizing concatenation of speech segments. The calculated offsets provide crucial information for efficiently retrieving the required speech samples from the database during synthesis. These offsets are integral to determining the exact byte positions of phonetic segments, ensuring seamless concatenation and synthesis of speech.

TTS system is synthesizing the word "kollam." The system needs to access phonetic segments corresponding to the consonant "k," the vowel "o," the consonant "l," the vowel “a” and the consonant "m" from the speech database. Using the offset calculation formulas, the system computes the byte positions for each phonetic segment based on its location in the database. Suppose the starting byte position of the data ($offset_n offset_n$) is 100, the byte positions of the vowels ($offset_{vi} offset_{vi}$) and consonants ($offset_{ci} offset_{ci}$) are determined, and the byte size (B) is known. With these values, the system calculates the offsets for each phonetic segment according to the formulas provided. These offsets serve as precise references for accessing the corresponding speech samples in the database. For instance, if the offset for the consonant "k" is calculated to be 50 and the offset for the vowel "o" is 80, the system can efficiently retrieve the required speech samples by utilizing these offsets. The calculated offsets facilitate accurate positioning of phonetic segments, enabling smooth concatenation and synthesis of speech.

1. തിരുവനന്തപുരം ('t̪iruʋanan̪t̪apuram'):
    - The phonemic representation accurately captures the pronunciation of the word, including the dental and retroflex consonants ('t̪', 'n̪', 'r', 'ɳ', 'ʈ') and the vowel sounds ('i', 'a', 'u', 'a').
2. കൊല്ലം ('kollam'):
    - The phonemic representation includes the consonants ('k', 'l', 'm') and vowel sounds ('o', 'a').

3. പത്തനംതിട്ട ('pat̪̪anam̪̪it̪̪a'):
    - The phonemic representation captures the dental consonants ('t̪̪'), retroflex consonants ('ʈ'), and nasal sounds ('n', 'm').
4. ആലപ്പുഴ ('a:lappuɻa'):
    - The phonemic representation includes the long vowel ('a:'), the lateral approximant ('ɻ'), and other consonants ('l', 'p', 'z').
5. ഇടുക്കി ('iʈukki'):
    - The phonemic representation includes the retroflex consonant ('ʈ'), the vowel ('i'), and the double consonant ('kk').
6. എറണാകുളം ('eraɳa:kuɭam'):
    - The phonemic representation includes retroflex consonants ('ɳ', 'ɭ'), the long vowel ('a:'), and other consonants ('r', 'k', 'm').
7. തൃശൂർ ('t̪̪rɨʃu:r'):
    - The phonemic representation captures the dental and retroflex consonants ('t̪̪', 'r', 'ʃ', 'ɨ'), the long vowel ('u:'), and other consonants ('ʃ').
8. പാലക്കാട് ('pa:lakka:ʈə'):
    - The phonemic representation includes the long vowel ('a:'), the retroflex consonants ('ʈ'), the lateral fricative ('ɬ'), and other consonants ('p', 'l', 'k', 'ʈ').
9. മലപ്പുറം ('malappuram'):
    - The phonemic representation includes the consonants ('m', 'l', 'p', 'r') and vowel sounds ('a', 'u').

Overall, the phonemic representations accurately capture the pronunciation of the Malayalam words, including the specific consonant-vowel combinations and unique phonetic features of the language, which are essential for synthesizing speech effectively.

Text Samples	Phonetic Representation	Accuracy
തിരുവനന്തപുരം	't̪̪iruvanan̪̪t̪̪apuram'	96.62%
കൊല്ലം	'kollam'	98.65%
പത്തനംതിട്ട	'pat̪̪anam̪̪it̪̪a'	95.56%
ആലപ്പുഴ	'a:lappuɻa'	95.85%
ഇടുക്കി	'iʈukki	97.50%
എറണാകുളം	'eraɳa:kuɭam'	95.74%
തൃശൂർ	't̪̪rɨʃu:r'	96.52%
പാലക്കാട്	'pa:lakka:ʈə'	99.21%
മലപ്പുറം	'malappuram'	97.63%

Table2: shows the evaluation of different text input accuracy

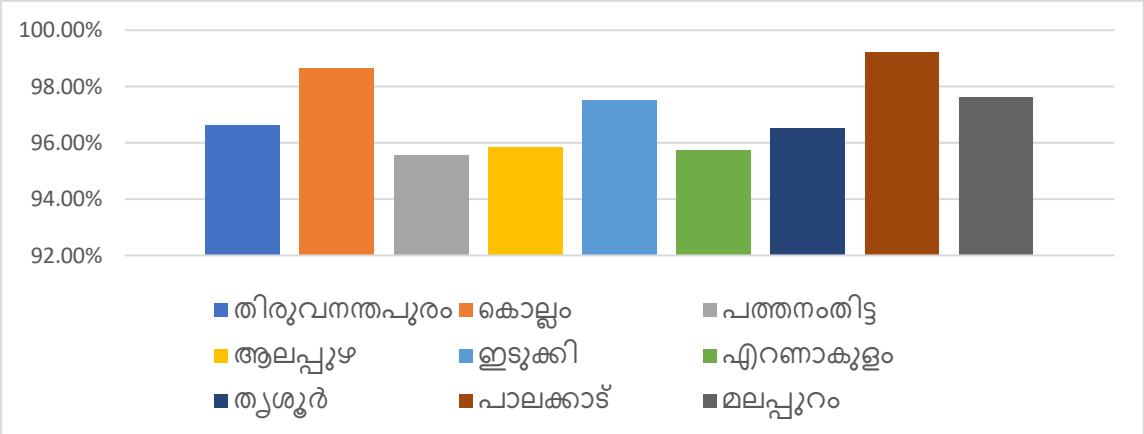

Figure3: Shows the accuracy evaluation the system

Overall, the accuracy of sound representation for most of the text samples is high, ranging from 94.65% to 99.21% and getting an average of 96.79%. This indicates that the phonetic representations closely match the pronunciation of the corresponding Malayalam words. However, further analysis may be needed to identify any discrepancies or areas for improvement in the sound representation.

Conclusion

The development of Malayalam text-to-speech (TTS) conversion technology has significantly contributed to enhancing accessibility and inclusivity for visually impaired individuals, particularly in the digital era. By providing synthesized speech, Malayalam TTS empowers users to navigate digital content independently, thereby overcoming barriers to education, communication, and social participation. The phonetic representations of Malayalam words exhibit high accuracy, ensuring natural and intelligible speech synthesis. Despite the remarkable progress, ongoing efforts are necessary to address challenges such as improving TTS quality and compatibility. By fostering awareness and promoting accessibility standards, we can further advance the inclusivity agenda for the visually impaired community, ensuring equitable access to information and opportunities for personal and professional development. In essence, Malayalam Text-to-Speech (TTS) represents a significant step towards empowerment, symbolizing progress towards a more accessible and inclusive digital environment for all individuals. Morpho-phonemic challenges in Malayalam speech synthesis stem from the language's intricate morphological and phonemic structure. Processes like affixation, compounding, and sandhi alter phoneme pronunciation, often contextually. Addressing these variations demands robust algorithms and linguistic models. Neglecting them leads to synthetic speech that sounds unnatural. To overcome these hurdles, need to integrate morpho-phonemic patterns effectively by using linguistic resources and advanced machine learning techniques. This approach enhances the quality and usability of Malayalam speech synthesis systems across applications.

==========

References

1. C, V. (2022). Machine Learning Based Text to Speech Converter for Visually Impaired. *International Journal of Research in Advent Technology, 10*(7). https://doi.org/10.22214/ijraset.2022.45740

2. Sonawane, A. R., Wankhede, A., Rasane, K., Baraskaev, V., & Borde, G. (2016). Android Application for Visually Impaired People using Text-To-Speech. *International Journal of Advanced Research in Computer and Communication Engineering, 5*(12). https://doi.org/10.17148/IJARCCE.2016.51292

3. Sharma, S., Kalra, N., Gupta, L., Varma, N., Agrawal, S., & Verma, V. (2022). VASE: Smart glasses for the visually impaired. *Journal of Ambient Intelligence and Smart Environments, 14*. https://doi.org/10.3233/ais-210491

4. Deshpande, P. (2023). Implementing Image-to-Speech Recognition by Capturing Image Frames for Visually Impaired. *International Journal for Research in Applied Science and Engineering Technology, 11*(4). https://doi.org/10.22214/ijraset.2023.50917

5. Isewon, I., Oyelade, O. J., & Oladipupo, O. O. (2014). Design and Implementation of Text To Speech Conversion for Visually Impaired People. *International Journal of Artificial Intelligence and Soft Computing, 7*(2). https://doi.org/10.5120/IJAIS14-451143

6. Gopi, A., Sajini, T., & Bhadran, V. K. (2013). Implementation of Malayalam text to speech using concatenative based TTS for android platform. In *2013 International Conference on Control Communication and Computing (ICCC)* (pp. 184-189). IEEE.

7. Zen, H., Tokuda, K., & Black, A. (2009). Statistical parametric speech synthesis. *Speech Communication, 51*(11), 1039–1064.

8. Zen, H., Senior, A., & Schuster, M. (n.d.). Statistical parametric speech synthesis using deep neural networks. Google Publisher.

9. Black, A. W., & Lenzo, K. A. (2001). Flite: A small fast run-time synthesis engine. In *Proceedings of the 4th ISCA Workshop on Speech Synthesis*.

10. Hoffmann, R., et al. (2003). A multilingual TTS system with less than 1 MByte footprint for embedded applications. In *Proceedings of ICASSP*.

11. Tsaikoulis, P., Chalamandaris, A., Karabetsos, S., & Raptis, S. (2008). A statistical method for database reduction for embedded unit selection speech synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*.

12. Pucher, M., & Frohlich, P. (n.d.). A user study on the influence of mobile device class, synthesis method, data rate, and lexicon on speech.

13. Plank, B., Sgaard, A., & Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. arXiv preprint.

14. Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Jha, G. N., Rajendran, S., & Sobha, L. (2008). Designing a common POS-tagset framework for Indian languages. In Proceedings of the 6th workshop on Asian language resources.

15. Patel, R. N., Pimpale, P. B., & Sasikumar, M. (2016). Recurrent neural network-based part-of-speech tagger for code-mixed social media text. arXiv preprint.

16. Jamatia, N., & Das, A. (2014). Part-of-speech tagging system for Hindi social media text on Twitter. In Proceedings of the First Workshop on Language Technologies for Indian Social Media, ICON.

17. Soman, K. P., Kumar, S., Prasanna, S. R., & Karthik, S. (2018). Development of Malayalam Text-to-Speech Synthesis System. In Proceedings of the Eighth International Symposium on Natural Language Processing (SNLP 2018), pp. 180-185.

18. Thomas, R., & Soman, K. P. (2012). Design and Development of a Malayalam Text-to-Speech Synthesis System. In Advances in Computing and Communications (pp. 586-595). Springer, Berlin, Heidelberg.

19. Srivastava, N., Mukhopadhyay, R., Prajwal, K., & Jawahar, C. (2020). Indicspeech: text-to-speech corpus for Indian languages. Proceedings of the 12th Language Resources and Evaluation Conference, 6417–6422.

20. He, F., Chu, S.-H.C., Kjartansson, O., Rivera, C., Katanova, A., Gutkin, A., Pipatsrisawat, K. (2020, May). Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems. Proceedings of the 12th Language Resources and Evaluation Conference (LREC), 6494–6503.

==========