Data and Metadata. 2023; 2:191
doi: 10.56294/dm2023191

ORIGINAL

A Grapheme to Phoneme Based Text to Speech Conversion Technique in Unicode Language

Una técnica de conversión de texto a voz basada en grafemas y fonemas en lenguaje Unicode

Chandamita Nath[1], Bhairab Sarma[1]

[1]University of Science & Technology, Meghalaya, Department of Computer Science, Meghalaya, India

Cite as: Nath C, Sarma B. A Grapheme to Phoneme Based Text to Speech Conversion Technique in Unicode Language. Data and Metadata. 2023; 2:191. https://doi.org/10.56294/dm2023191

Submitted: 13-08-2023 **Revised:** 09-11-2023 **Accepted:** 29-12-2023 **Published:** 30-12-2023

Editor: Prof. Dr. Javier González Argote

ABSTRACT

Text-to-speech conversion can be done with two approaches: dictionary-based (database) approach and grapheme-to-phoneme (G2P) mapping. One of the drawbacks of this approach is its performance depends on the size of the dictionary or database. In the case of domain specific conversion, a simple rule -based technique is used to play pre-recorded audio for each equivalent token. It is easy to design but its limitation is mapping with the sound database and availability of the audio file in the database. In general, grapheme to phoneme conversion can be used in any domain. Advantages are the limited size of the database required, ease of mapping and compliance with domain. However, G2P suffers from pronounce ambiguity (formation of audio output). This paper will discuss about the grapheme-to -phoneme mapping and its application in text to speech conversion system. In this work, Assamese *(an Indian scheduled Unicode language)* is used as the experimental language and its performance is analysis with another Unicode language (Hindi). English (ASCII) language will be used as a benchmark to compare with the target language.

Keywords: CNN; Grapheme to Phoneme; NLP; TTS Conversion; Unicode.

RESUMEN

La conversión de texto a voz puede realizarse con dos enfoques: el basado en diccionario (base de datos) y el de mapeo grafema-fonema (G2P). Uno de los inconvenientes de este método es que su rendimiento depende del tamaño del diccionario o de la base de datos. En el caso de la conversión específica de dominio, se utiliza una técnica simple basada en reglas para reproducir audio pregrabado para cada token equivalente. Es fácil de diseñar, pero su limitación es el mapeo con la base de datos de sonido y la disponibilidad del archivo de audio en la base de datos. En general, la conversión de grafema a fonema puede utilizarse en cualquier ámbito. Sus ventajas son el tamaño limitado de la base de datos necesaria, la facilidad de mapeo y la conformidad con el dominio. Sin embargo, G2P adolece de ambigüedad de pronunciación (formación de la salida de audio). Este artículo trata sobre el mapeo grafema-fonema y su aplicación en el sistema de conversión de texto a voz. En este trabajo, se utiliza el asamés (un idioma indio Unicode programado) como idioma experimental y se analiza su rendimiento con otro idioma Unicode (hindi). El inglés (ASCII) se utilizará como referencia para compararlo con el idioma de destino.

Palabras clave: CNN; grafema a fonema; PNL; conversión TTS; Unicode.

INTRODUCTION

Text to speech (TTS) conversion is an important activity in natural language processing (NLP). A written text is converted into sound with pronounce (acoustic) generation. TTS is important for reading a written text, like newspaper, articles, books, messages etc. With the help of this system, a visibly impaired person can also read text. Not only that, men at working like cooking, driving, travelling etc.can also read newspaper or written article. Text can be read with different acoustics (male or female voice) and with upper or lower tune.

Prior to the decade, word by word conversion techniques were widely used which we termed as dictionary-based approach. Here texts like speech, message, or words are taken as input in the system and the equivalent acoustic is search in the pre-defined recorded sound database.(1,2) and plays the audio upon successful mapping. For unsuccessful searching, the system either gets stuck or reads the word character by character. The performance of this approach varies domain to domain and on the volume of the dataset (as recorded sound). To increase the performance and make it a domain independent TTS system, a new approach has been used where instead of word, graphemes (smallest part of written text) are converted into phonemes (smallest part of a sound) which are termed as G2P conversion. In the proposed approach, the TTS system consists of an NLP module and a Digital Signal Processing (DSP) module. The NLP module converts the text i.e. the graphemes to a string of phonemes.(3,4) An encoder-decoder system is used to synchronize elements of graphemes with a bottle-neck approach. And then encodes the tone and prosodic information in the output audio stream. In a concatenative synthesis approach, the DSP module obtains the sound files from an acoustic inventory corresponding to the string of phonemes or diaphones and concatenates the outputs.(2) Finally, it modulates the sound according to the intonation and prosodic information. However final pruning is required to adjust the phonetic features to accomplish an intelligent speech. Some other linguistic features are also considered for analysis to develop the intelligibility of the speeches(3,5) like modulation, tone, gender etc.

Unicode languages are structurally difficult to pronounce based on their graphemes or decomposing their syllable. Similar words may produce different acoustics depending on concerned grammatical rules. Rules of pronunciation of words with vowel ending or consonant ending or contextual conformity will lead in complex design. This paper will include some important common rules of Unicode languages and create an appropriate dataset for machine training, test them under the supervision pre-assigned rules as per grammar of the target language. Assamese is an Indian language *(First in fifteenth scheduled languages)* spoken by the people of northeastern states of India, origin from Sanskrit, similar in structural conformity with Bengali, Odiya and Hindi.(6,7,8,9) These languages follow more or less same grammatical rules and phonetic features.

Previous development

In the previous work,(10,11,12) a dictionary based approached has been formulated and experimented with three different Indian languages.(12) As stated, the result of the system was not satisfactory because of inflectional behavior of Indian languages and limited sizes of applied databases. Another reason was the requirement of strong mapping techniques in accessing sound databases.

A rule based G2P technique has been describe in(3,11) and was implemented in Hindi *(Indian national language)* where some rules for omission and addition are applied in phonetics were explained.

Another rule-based approach proposed by Kumar and et.al.(3), described a rule-based approach in Hindi language with phonetically mapping synthesis. In this approach, a Devnagri grapheme is represented by a code that is either 2 or 3 bytes long. As per the rule all vowel modifiers and most pure consonants are encoded using 2 bytes. The two-byte codes cover most of the Hindi graphemes. Three-byte sequences represent the stand-alone vowels [a/अ, A/आ,i/इ, I/ई, e/ए, oi/ऐ, ou/औ], some consonants [C/ छ, ja/ज, xa/अ, ca/च, T /ट, D/ड] and "dandaviram" (*full stop*) [.]. Characters coded with Unicode are structurally different from ASCII code. Unlike the Roman script, the Devnagri scripts are not linear. Ligatures represent consonant clusters. Moreover, the script is not causal. The order of display of graphemes does not strictly represent the order of phonemes of the language.(8)

Grammatically, graphemes consist of all of the letters and letter combinations that represent phonemes of the language. Indian languages are coded with Unicode where a character can be represented by either a standalone vowel or vowel in combination with one or more consonants. A grapheme in UTF-8 is represented as a sequence of length 1 to 6 bytes. English (coded with ASCII) characters are represented as a single byte. A Devnagri or Assamese grapheme is represented by a code that is either 2 or 3 bytes long.(7,8)

In this current work, graphemes are decomposed from the input chunk of text encoded with Assamese font *(Unicode, Azhagi+)* and then sequence to sequence G2P mapping will be executed to produce sound with proper matching technique.

Grapheme to Phoneme (G2P) conversion

There are basically three broad categories of Grapheme to Phoneme (G2P) conversion.(7) They are-Phonological Rule Based Approach, Data Driven Approach and Statistical Approach.

Phonological rule- based approach is based on the assumption that the pronunciation of a letter or letter substring can be found if enough information is known about its context. It uses handwritten phonological rules of the form A[B]C → d. This states that the letter substring B, in the context of A and C, rewrites as phoneme d. Since the mapping from orthography to sound is complex, especially for English, more than one rule is typically needed for the transformation. As described in[9], the conflicts that occur between rules as they are applied and resolved by keeping them in a set of sub-lists grouped by initial letter and ordered by the specificity of the rule. The most specific rule is placed at the top with more general rules towards the bottom. During transformation, the sub-lists are searched in a left to right process; for each letter, the appropriate sub-list is searched from top to bottom until a match is found.

Datadriven approach as explained in[11] uses a model which maps graphemes to phonemes following automatic process by processing a training data set. The operations of many data driven approaches are not different from the phonological rules; the questions stored in a decision tree could be reformulated as context-sensitive rules. In this approach, a letter to phoneme alignment is created automatically from the lexical database. The input word is then scanned for substrings that match the alignment database, and a pronunciation stream is created. The decision function finds a possible pronunciation for the word by traversing the pattern and concatenating the phoneme labels on the arcs. The path chosen is shortest, and in the case of a tie, a heuristic is used.

Statistical Approach uses a statistical method to train a model from the data set *(Markovian Approach)*. The statistical methods mostly use Hidden Markov Model for G2P conversion whereas Data driven approach merely used CNN model. In CNN approach, different layers represented different levels of activities. Decomposition of words into grapheme and then further decomposition of secondary forms of composite characters *('yuktaakshar')* is done in the consequent layer.

Grapheme Decomposition in Unicode

Graphemes are the smallest part of a written text. In English each alphabet is a grapheme and their individual pronounce is called as phoneme. Upon decomposition of word in Indian language, a number of graphemes may come out whenever they are converted to phoneme, the actual pronounce of the word would not be produced due its grammatical structure of the language.

In this paper, symbolically graphemes are often notated as <a>, <b> etc. whereas phonemes are notated as /a/, /b/ etc. and the phonetic transcriptions will be notated as [a], [b]. With this syntax, Grapheme analysis of an input Assamese sentence is explained in table 1.

Table 1. Grapheme analysis of Assamese text

Input sentence	ব্ৰহ্মপুত্ৰঅসমৰএখনবৃহৎনদী / 'brahmaputraasamaraekhanbruhat` nadI'
Grapheme	<ব><ৰ><হ><ম><প><উ><ত><ৰ><এ><খ><ন><অ><স><ম><ৰ><ব><ঋ><হ><ৎ><ন><দ><ঈ>
Phoneme:	/ba/ /ra/ /ha/ /ma/ /pa/ /u/ /^t/ /ra//—/ /a/ /xa/ /ma/ /r/ /—/ /ae/ /xn/ /brue/ /ha/ /~t/ /na/ /dE/
Phonemes acoustic	/bRah/ /ma/ /puT/ /ra/ /-/ /a/ /xa/ /ma/ /r/ /eKn/ /brul/ /H//~t/ /Na/ /dE/
Assamesesound mapping	/ব্ৰহ / /ম/ /পুত/ /ৰ/ /-/ /অ/ /স/ /ম/ /~ৰ//-/ / এ/ /খ/ /~ন//-/ /ব্ৰী/ /হ/ /~ত/ /-//ন// অ//দী/

Phoneme /-/ is used to represent gap*(silent)* between two phonemes produced by whitespace in input sentences. On conversion into acoustic sequence, it can be listen as/barahama-p-u-ta-ra/ or /brah-ma-put-ra/, where an impression is found after /brah/. This impression has been produced during grapheme decompositions. In our experiments, we used encoder-decoder architectures including one Supervised Approach. In this structure, the encoder computes a representation of each input sequence, which is a text *(Grapheme)* and the decoder generates an output sequence which is *.wav* format *(Phoneme)* based on the learned representation. The block diagram of this model is given in figure 1.

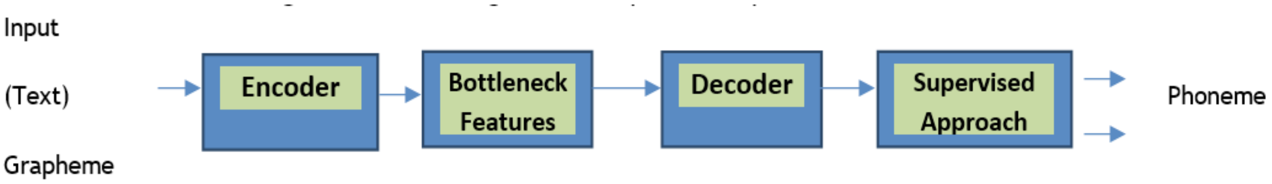

Figure 1. Block diagram of Grapheme to phoneme conversion

Encoder- Decoder

The main idea of the Encoder- Decoder approaches have two steps: the first step is mapping the input sequence to a vector; the second step is to generate the output sequence based on the learned vector representation. Encoder-decoder models generate an output after the complete input sequence is processed

https://doi.org/10.56294/dm2023191

by the encoder, which enables the decoder to learn from any part of the input without being limited to fixed context windows. The decoded grapheme from an input text is input to phoneme generation phase and each phoneme is enter to an encoder. The encoder encoded the output phonemes to produce continuous audio using bottleneck features. Morphological dissemination the Assamese word[kitaap]“কিতাপ” is explained in the figure 2. The initial phase of our system has been coded with PHP, and phonetic syntheses were experimented in ‘FL Studio 20’*(an audio mixing software)*. In level 1, each character of the word decomposed with an iteration. Here, the output is grapheme of each character disseminating punctuation markers.

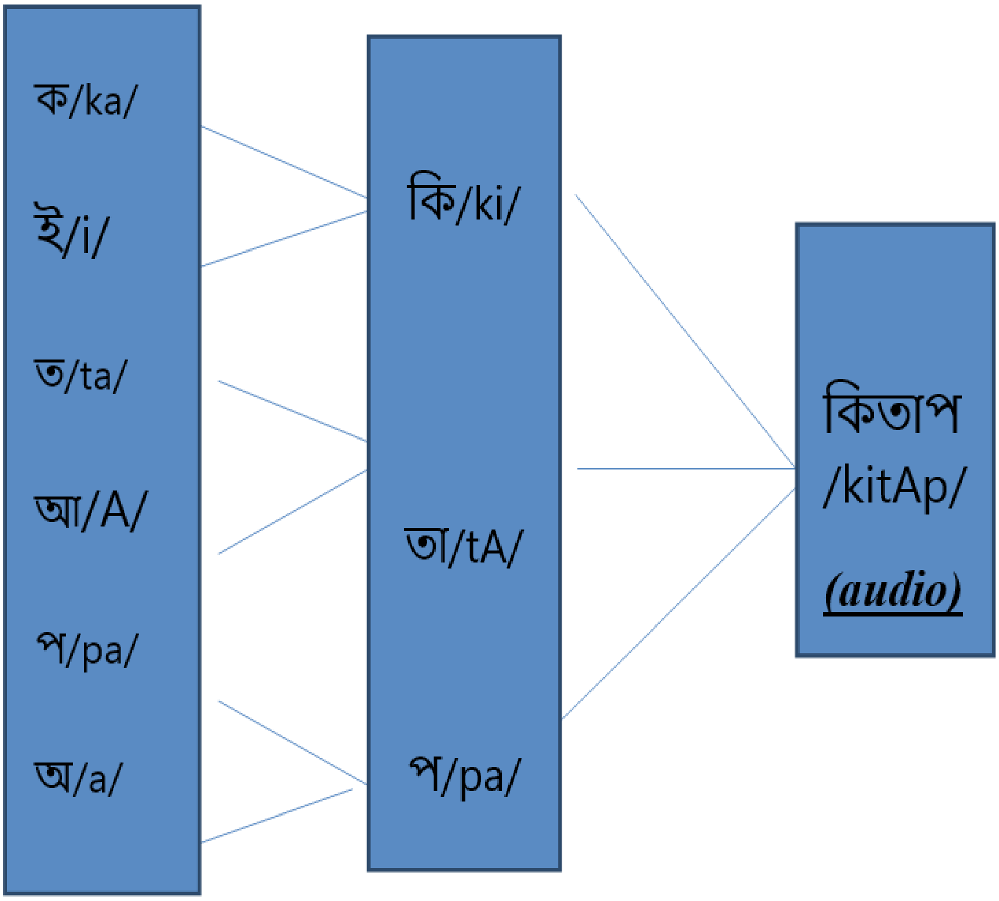

Figure 2. Morphological dissemination of Assamese word

Rule Implementation

The accuracy level of this TTS degraded in Assamese language because of different inflections rules *(composition rule)*. Few considerable examples are:

i. When consonants are added with a vowel, the consonant character takes a secondary form with a ‘matraa’ of the added vowel which again treated as a single character.

ক + ই→কি ক+ উ→কু ত+আ→তা ৰ +ই→ৰি etc.

ii. When a consonantareadded with anotherconsonant, it becomes a compound character, merely called ‘*yuktakshar’* further treated as a single character.

ক+ক→ক্ক ৰ + ত→ৰ্ত ত+ৰ→ত্ৰ ট + ট→ট্ট etc.

iii. Therefore, the number of characters in ‘ব্ৰহ্মপুত্ৰ’ is four instead of 10 independent graphemes. and the number of characters in ‘নম্ৰতা’ is three instead of five. The graphemes are /na/ /ma/ /ra/ / ta/ /aa/ which needs five independent phonemes.

Decomposition problems

Words are separated from a sentence using tokenization. After tokenization each unique word is decomposed by a looping structure to get the character and then the graphemes. Single characters can be extracted in a single iteration; however, for complex characters it may require multiple iterations. Composition-decomposition operations are used to find the actual graphemes. These problems arise due to their inflection nature. A few observations are outlined in table 2.

RESULT ANALYSIS

Table 2. Composition-decomposition ambiguity with inflections for different secondary forms of vowel addition

Inflections	Composition	Decomposition
Consonant with Vowel	K+ A->kA (ক + আ→কা) K+I->kI(ক + ঈ→কী)	kA->k+…. (not a valid character, secondary form) kI->k + …. (not a valid)
Consonant to Consonant	k+k->kk (ক + ক→ক্ক) k+l- kl(ক + ল→ক্ল)	kallol- k + l+ l+o+l /(কল্লে◌াল→ক +ল + ল + ** +ল (not valid)
Note: ** symbolic representation of vowel		

Our previous work was concluded in sound database and its application in TTS.(12) We presented a sound database which consisted of 4000 different unique words with their equivalent audio. This corpus was created from different sources like Assamese newspapers, articles and manually typed words from a standard Assamese language. Among those 4000 unique words, approximately 450 audios have been created. The program has been developed in HTML code to read words as an input and mapped *(indexed table created)* it in the dictionary. If the input word is found matching exactly in the sound database, the equivalent audio file will be played for a while. We tested this system with different input word of three different languages and found around 30 percent accuracy in Indian languages where the system claimed 77 percent accuracy in case of English language. The system was demonstrated with three different Unicode languages and compared its performance with English language keeping standard size of database. The results are shown in table 3.

Table 3. Accuracy comparison of correctly spelled word searching in the sound database with single word input in ASCII vs Unicode languages

Experimented Language	Number of entries in the database	Number of word input	Spelled Correctly	Accuracy PC
English	412	88	66	70 %
Hindi	334	56	18	32 %
Assamese	450	197	84	42 %
Bengali	243	68	19	29 %

The variations in accuracy are observed because of Indian languages are more inflectional in nature. Secondly, the size of the dictionary is also another important factor. The higher the population results better accuracy as expected.

Next with the CNN second approach, we tested our system with same database having input as a plain text file consist of approximately equal number of words. Performance of second succession is depicted in table 4.

Table 4. Accuracy with CNN approach

Experimented Language	Number of words in input text	Spelled Correctly	Accuracy PC
English	112	62	56 %
Hindi	140	55	40 %
Assamese	163	58	36 %
Bengali	98	31	32 %

Ultimately, it has been revealed that Assamese language accuracy level displays somewhat higher with G2P in comparison to existing CNN system based on their performances. In the case of English language, performance of G2P is greatly improved (figure 3). From these experimental results, precision have been calculated for cases as:

Precision result claimed that G2P performance is better than DB (comply with Alok Parliker et al., 2016) in English and Assamese language where is contradicted in Hindi and Bengali language. One reason may be volume (sample size) the dictionary in DB approach. Finally, since our target language is Assamese TTS, we superimposed G2P in Assamese language, but a considerable factor is that all the three languages are coded with Unicode. On the other hand, in G2P the limited size of the pre-recorded voice database makes the system simpler and faster access of the database.

$$precision = \frac{Number\ of\ words\ Spell\ correctly}{Correct\ words(true\ Positive)in\ the\ dictionary + spell\ wrongly(true\ Negative)}$$

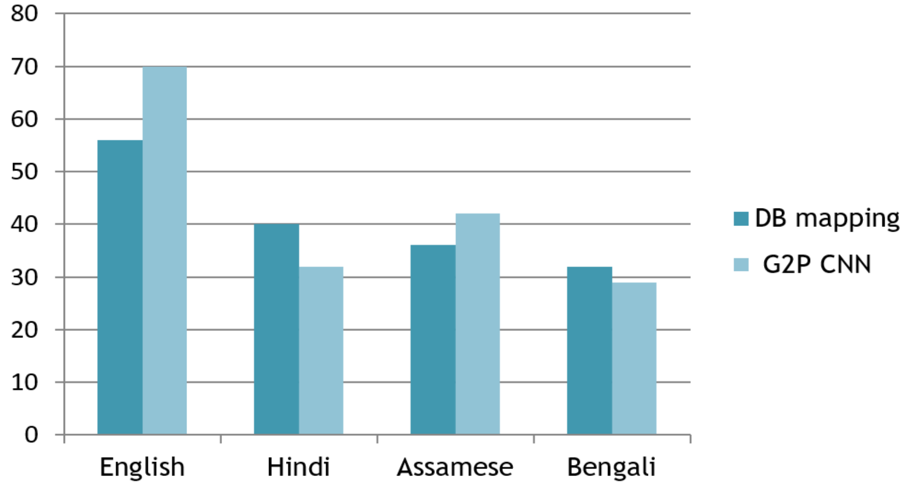

Figure 3. Precision comparison between DB vs G2P approach with three languages

CONCLUSION

The goal of this work was to create an automatic Assamese text-to-speech converter system. As English and other Indian languages have previously paved the road for this voyage, our aim is to apply in Assamese as well, which has seen relatively little work in the literature review. We created a few systems (dictionary-based, DSP-based, sound database, and smart tokenizes) utilizing various methodologies, and in the end, the G2P approach proved to be quite effective. Research on alternative methods and their application to many languages has been continuing. Different ligatures and annotations are used in Unicode languages. An effective TTS system should also take into account other factors like vocal power and tune Synthesis. In the future, there might be a complete Assamese language ATTS version available on the Internet with consideration of all grammatical rules and features.

ACKNOWLEDGMENT

Authors are thankful to the publishers for giving us the opportunity to share our research work. We are grateful to those without whose support; we could not concrete the new concepts in this field. Not only that, but reviewer's comments will also help us in a great manner. We always remember their valuable suggestions. Our sincere thanks go to all colleagues of the Department of Computer Science of USTM for their continuous inspiration.

REFERENCES

1. Arora A, Gessler Luke, Schneider N, (2020), Supervised Grapheme-to-Phoneme Conversion of Orthographic Schwas in Hindi and Punjabi, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7791–7795.

2. Nath C, Sarma B. (2023), Analysis of Inflectional Behavior in Indian Languages using Features Extraction Techniques, 2023 International Conference on Advancement in Computation & Computer Technique, IEEE, 8 June 2023, DOI: 1109/InCACCT57535.2023.10141783

3. Alok Parlikar, Sunayana Sitaram, Andrew Wilkinson and Alan W Black (2016), The Festvox Indic Frontend for Grapheme-to-Phoneme Conversion, Carnegie Mellon University Pittsburgh, USA, https://www.cs.cmu.edu/~awb/papers/LREC16_parlikar.pdf, WILDRE3, W3RD WORKSHOP ON Indian language data: resources and evaluation.

4. Kumar C.S.,Govind.D.Menon, Nijil Chalil, Sethunath R. and Narwaria M (2006), Grapheme to phone conversion for Hindi, Conference on Oroiiental COCOSDA, 2006,Amrita Vishwa Vidyapeetham, Ettimadai,

Coimbatore, Tamil Nadu, INDIA.

5. Srikanth Ronanki, Siva Reddy, BajibabuBollepalli (2016), DNN-based Speech Synthesis for Indian Languages from ASCII text , 9th ISCA Speech Synthesis Workshop, September 2016, DOI: 10.21437/SSW.2016-12

6. Caero L, Libertelli J. Relationship between Vigorexia, steroid use, and recreational bodybuilding practice and the effects of the closure of training centers due to the Covid-19 pandemic in young people in Argentina. AG Salud 2023;1:18-18.

7. Ogolodom MP, Ochong AD, Egop EB, Jeremiah CU, Madume AK, Nyenke CU, et al. Knowledge and perception of healthcare workers towards the adoption of artificial intelligence in healthcare service delivery in Nigeria. AG Salud 2023;1:16-16.

8. Mousmi A. (2016), Grapheme-to-phoneme conversion scheme for sentence-by-sentence learning of korean manuscript using joint sequence statistical model, International journal of current engineering and scientific research (ijcesr) issn (print): 2393-8374, (online): 2394-0697, volume-3, issue-7, 2016, DOI:10.21276/ijcesr

9. Singh A. K. (2016), A Computational Phonetic Model for Indian Language Scripts, Language Technologies Research Centre IIIT, Hyderabad, India. http://cdn.iiit.ac.in/cdn/ltrc.iiit.ac.in/anil/papers/cpms-long-iwlc-06.pdf

10. Pathak N, Talukdar P. H., (2013), The Basic Grapheme to Phoneme (G2P) Rules for Bodo Language, International Journal of Computing, Communications and Networking, Available Online at http://warse.org/pdfs/2013/ijccn06212013.pdf

11. Aliya Deri, Knight K., (2016), Grapheme-to-Phoneme Models for (Almost) Any Language, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 399–408, Berlin, Germany, August 7-12, 2016. c 2016 Association for Computational Linguistics.

12. Chourasia V, Samudravijaya K (2005), Phonetically Rich Hindi Sentence Corpus for Creation of Speech Database, Available at: mc.iet@dauniv.ac.in. https://www.iitg.ac.in/clst/visitors/samudravijaya/publ/05phoneticallyRichSentHindi.pdf

13. Magdum D, Patil T, Suman M., (2019), Schwa Deletion in Hindi Language Speech Synthesis, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-6S, August 2019

14. Auza-Santivañez JC, Lopez-Quispe AG, Carías A, Huanca BA, Remón AS, Condo-Gutierrez AR, et al. Improvements in functionality and quality of life after aquatic therapy in stroke survivors. AG Salud 2023;1:15-15.

15. Castillo-González W. Kinesthetic treatment on stiffness, quality of life and functional independence in patients with rheumatoid arthritis. AG Salud 2023;1:20-20.

16. Yolchuyeva S., Géza Németh and Bálint Gyires-Tóth (2019), Grapheme-to-Phoneme Conversion with Convolutional Neural Networks, Application Science; Published: 18 March 2019, Appl. Sci. 2019, 9, 1143; DOI:10.3390/app9061143 available at: www.mdpi.com/journal/applsci

17. Nath C., Sarma B., (2021), A New Concept of Sound Database for Development of Spelling Generator, Journal of Biological Engineering Research and Review, 2021; 8(2): 01-425 ISSN: 2349-3232, Conference Proceeding ADVANCEMENT IN ARTIFICIAL INTELLIGENCE THEORIES AND APPLICATIONS IN BIOMEDICAL ENGINEERING, NIT Patna, 2021

18. Choudhury M (2003), Rule Based Grapheme to Phoneme Mapping for Hindi Speech Synthesis, Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur, Corpus ID: 2392234https://play.ht/text-to-speech-voices/indian-hindi/

19. Basu J, Basu T, Mitra M, Mandal SKD (2009), Grapheme to phoneme (G2P) conversion for Bangla. In: 2009 Oriental COCOSDA international conference on speech database and assessments, Urumqi, China, Aug 2009, pp. 66–71, doi.org/10.1109/ICSDA.2009.5278373

20. Badhon S, Rahaman MdH, Rupon FR, Abujar S (2020), State of art research in Bengali speech recognition. In: 2020 11thInternational conference on computing, communication and networking technologies (ICCCNT), Kharagpur, India, July 2020, pp. 1–6.

FINANCING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST

Not Applicable.

AUTHORSHIP CONTRIBUTION

Conceptualization: Chandamita Nath, Bhairab Sarma.
Data curation: Chandamita Nath, Bhairab Sarma.
Formal analysis: Chandamita Nath, Bhairab Sarma.
Research: Chandamita Nath, Bhairab Sarma.
Methodology: Chandamita Nath, Bhairab Sarma.
Supervision: Bhairab Sarma.
Drafting - original draft: Chandamita Nath, Bhairab Sarma.
Writing - proofreading and editing: Chandamita Nath, Bhairab Sarma.