I^{2}R Text-to-Speech System for Blizzard Challenge 2013

S. W. Lee [1], *Minghui Dong* [1], *Shen Ting Ang* [1] *and Min Min Chew* [2]

[1] Human Language Technology Department, Institute for Infocomm Research,
A*STAR, Singapore 138632
[2] Nanyang Technological University, Singapore 639798

{swylee, mhdong}@i2r.a-star.edu.sg, angshenting@gmail.com, mmchew1@e.ntu.edu.sg

Abstract

This paper describes I2R's submission to the Blizzard Challenge 2013 speech synthesis evaluation. There are two sections this year: an English audio book section, and an Indic language speech synthesis section. Unit-selection was used for the English task; while HMM-based TTS was used for the Indian tasks. The Indian languages each have some distinct properties which need to be accounted for in the synthesis process. The Indian TTS system achieved a high level of intelligibility in the listening tests conducted.

Index Terms: speech synthesis, unit selection, HMM-TTS, Indian, English, Tamil, Hindi, Bengali, Kannada

1. Introduction

This year's challenge is divided into two sections – an English audio book section and a section on synthesis of Indic languages.

In recent years, there has been a trend of increased market demand for audio books. In addition to being useful for the visually-impaired, the audio book has also become a popular mode of entertainment for commuters. It is common to see popular books being sold as audio books on web retailers such as Amazon; Amazon's electronic book reader, the Kindle, also has built-in support for audio books. This increased popularity means a huge potential for application of text-to-speech (TTS) systems to the production of audio books.

India has one of the world's largest national populations and is home to many languages. This year's task picked out the four most widely-spoken languages for text-to-speech synthesis. The reference texts given are in the script for each of the original languages; one of the challenges is to accurately transliterate this into a sequence of phonemes which are used in existing TTS systems.

The rest of the paper is organized as follows: Section 2 covers the English audio book task; Section 3 covers the text-to-speech synthesis system for the four Indian languages, while our concluding statements are given in Section 4.

2. English

There are two tasks for English synthesis in this year's Blizzard challenge. Task EH1 is to build synthesis system from provided unsegmented audio data. Task EH2 is to build synthesis system based on segmented data. Due to time constraints, we only took part in task EH2. The wave files for EH2 were extracted from the audio books of Black Beauty and Mansfield Park. There were a total of 9733 valid speech files.

This year, the I^{2}R entry for English synthesis adopted the unit-selection based approach. The basic unit is the phone-sized unit. The wave files are labeled automatically by using forced alignment with HTK.

2.1. The acoustic and prosodic parameters

For each unit, we first calculated a set of parameters that describes the spectral and prosodic features of each phone as well as its frame boundaries. These parameters are chosen to include all the possible parameters in our consideration. The main values that we capture include the statistical values of each phone as well as the values of boundary (start and end) frames of the unit. The initial parameter set that we used consists of spectral features (MFCC), pitch features, duration features, and energy features. The parameter set forms a long vector, which contains a lot of redundancy. We used principal component analysis approach to reduce the number of dimensions of the data set. The dimensionally-reduced vector is considered a compact form of representation of the prosodic and spectral features of the unit. Finally, we have a 40-dimensional vector.

Though the acoustic parameters cover both spectral and prosodic information, we still need a set of prosodic parameters to emphasize the prosodic properties in speech. The prosodic parameters for each unit consist of pitch mean, duration, energy of the units.

2.2. Linguistic features

Linguistic features are derived from the input text. They are used for predicting the acoustic and prosodic parameters. We used the label files received with the data to generate linguistic features. We have derived the following linguistic features: current and context units, syllable level information, word level information, and utterance level information. Putting all the features together, we form an input linguistic feature vector of 30 elements.

2.3. Parameter prediction

The acoustic parameter prediction process calculates the parameters from the linguistic features. In our system, the linguistic features are the predictors and the acoustic and prosodic parameters are the responses. We built our models using the CART approach. Each individual parameter is predicted separately with a CART tree.

2.4. Unit selection

The unit-selection process is based on a cost function that consists of two parts (a) a target cost to measure the difference between the target unit and the candidate unit, (b) a joint cost to measure the acoustic smoothness between the concatenated units.

Our target cost further consists of three parts (a) the cost of acoustic parameters, (b) the cost of prosodic parameters, and (c) the cost of context linguistic features. The target cost c_t is defined as the following:

$$c_t = w_{ta}c_{ta} + w_{tp}c_{tp} + w_{tl}c_{tl} \qquad (1)$$

where c_{ta} , c_{tp} and c_{tl} are the cost of acoustic parameters, prosodic parameters and linguistic features respectively, and w_{ta}, w_{tp} and w_{tl} represent their corresponding weights.

The cost of the linguistic feature is to ensure the general spectral and prosodic accuracy of the candidate unit. Units with wrong pronunciation labels, which are generated due to grapheme-to-pronunciation mistakes, can also be eliminated by linguistic cost. However, due to the varsity of speech, using linguistic cost on its own may lead to extreme cases of abnormal spectrum and prosody too easily. The use of cost of acoustic parameters can avoid the selection of such extreme cases, because statistical models favor average values. The use of prosodic cost is to emphasize the importance of prosodic features.

3. Indian Languages

In addition to the English task described above, there is an Indian task (2013-IH1) which involves synthesizing voices for four Indic languages. These languages are Hindi, Bengali, Kannada and Tamil. It is a pilot phase of the 2014 challenge, which will probably include a few other Indian languages.

The speech data are extracts from the IIIT-H Indic speech databases [1]. For each of the four languages, 1000 sentences of speech data were collected from a native, non-professional speaker in quiet office environments. This approximates to one hour. These data are in 16 kHz sampling rate. The corresponding text is written in Unicode UTF-8 format [2]. In our implementation, the segment labels in [1] are applied. Two types of sentences are required to generate in the 2013-IH1 task, specifically, WPD sentences and SUS sentences. WPD sentences are public domain text available from Wikipedia pages. SUS sentences are semantically unpredictable ones. 100 sentences were generated for each type.

Voices for the four Indian languages were individually built. In the following, system designs, such as phone sets, grapheme-to-phoneme (G2P) mappings, training, etc., and the generation of testing sentences will be given in details.

3.1. System design

Among the many languages of India, Hindi, Bengali, Kannada and Tamil are commonly spoken. Most Indian languages have phonemic orthographies. This means their graphemes correspond to their phonemes. Consequently, G2P rules are applicable for synthesis [1]. Nevertheless, there is neither pronunciation dictionary nor readily available G2P rule for the four languages [1]. We learned the G2P rules based on these IIIT-H databases.

Hidden Markov model-based text-to-speech (HMM-TTS) framework [3] was adopted here for all the four languages, based on the speaker-dependent training demo released in [4].

3.2. Writing systems, phone set and G2P rules

While some Indian languages have own writing systems; some others share the same writing system called Devanagari. For example, Bengali, Kannada and Tamil have own sets of alphabets. Devanagari is used in Hindi. These Indian text inputs stored in UTF-8 format will be converted to phonetic representations. As this conversion is subject to the properties of the specific writing system, this section will highlight the properties involved. The phonemic orthographies of most Indian languages further lead to direct relationships between the writing systems and the associated phone sets. Hence, these phone sets will be given along.

3.2.1. Hindi writing system

The writing system of Hindi, Devanagari, is an abugida or alphasyllabary, which means it is a segmental writing system based on symbols, primarily consonants. Vowels are secondary as they are mostly seen as diacritics combined with consonants. There are twelve vowels and 33 main consonants in Devanagari. Depending on the role of a given vowel, it may be in independent form or dependent form, where these two forms have different appearances. When a vowel is used at the beginning of a word or following another vowel, the independent form is used. When it follows a consonant, the dependent form is used. Figure 1 lists the Hindi vowels in independent form and dependent form. The twelfth vowel (अ) does not have a dependent form.

आ इ ई उ ऊ ऋ ए ऐ ऑ ओ औ

◌ा ◌ि ◌ी ◌ु ◌ू ◌ृ ◌े ◌ै ◌ॉ ◌ो ◌ौ

Figure 1: *Vowels for Hindi, top row: independent vowel, bottom row: corresponding dependent vowels*

There are 33 consonants in Hindi. Each of them has an inherent schwa vowel "a", which can be changed to another one. Take an example, when the consonant "ka" changes to "ki", the corresponding Devanagari text changes, as shown in Fig. 2.

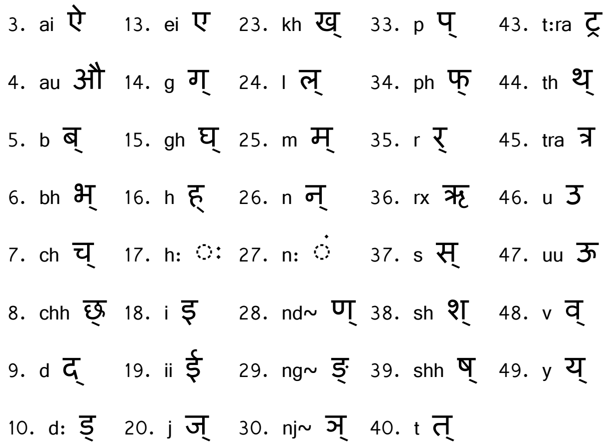
क ka → कि ki

Figure 2: *How Devanagari text changes from inherent vowel "a" to another vowel, with the use of dependent vowel form*

The Indian texts in IIIT-H databases are encoded in UTF-8 format. Each alphabet is represented by a unique Unicode, which is a group of four hexadecimal values. By decomposing the Unicode-formatted Indian texts into a sequence of consonants and independent vowels, this sequence essentially represents the phone sequence for this utterance. This decomposition has been applied for the four Indian languages.

Our phone set for Hindi is based on the segment labels in [1] and the above properties of Hindi. Fig. 3 gives the 49 phones we used in our Hindi system (The meaning of adjacent Hindi alphabets will be explained in later section). There are another two phones SIL and ssil, which are silences. SIL is a long silence while ssil is a short silence.

1.	a	अ	11.	d:h	ढ्	21.	jh	झ्	31.	oo	ओ	41.	t:	ट्
2.	aa	आ	12.	dh	ध्	22.	k	क्	32.	o~	ऑ	42.	t:h	ठ्

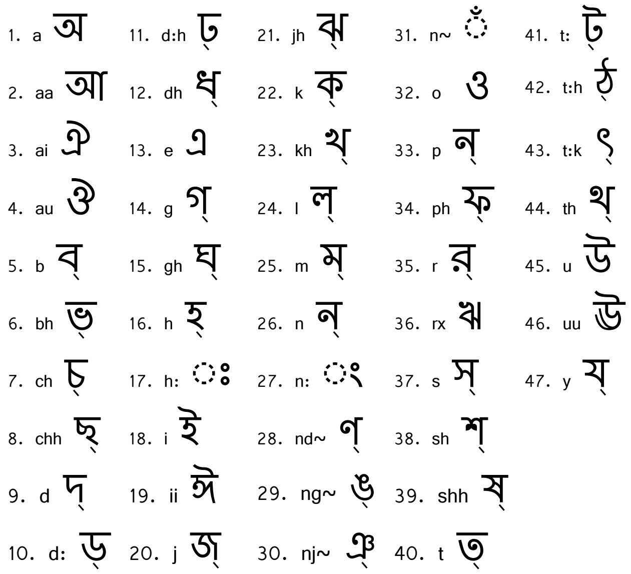


3. ai ऐ	13. ei ए	23. kh ख्	33. p प्	43. t:ra ट्र
4. au औ	14. g ग्	24. l ल्	34. ph फ्	44. th थ्
5. b ब्	15. gh घ्	25. m म्	35. r र्	45. tra त्र
6. bh भ्	16. h ह्	26. n न्	36. rx ऋ	46. u उ
7. ch च्	17. h: ः	27. n: ं	37. s स्	47. uu ऊ
8. chh छ्	18. i इ	28. nd~ ण्	38. sh श्	48. v व्
9. d द्	19. ii ई	29. ng~ ङ्	39. shh ष्	49. y य्
10. d: ड्	20. j ज्	30. nj~ ञ्	40. t त्	

Figure 3: *49 Hindi phones*

3.2.2. *Bengali writing system*

Bengali has its own writing system, although it is closely similar to Devanagari [5]. There are 36 consonants and eleven vowels. The Bengali script is also an abugida. There are 47 distinct phones (excluding SIL and ssil). Similarly, they were obtained from the IIIT-H database. Fig. 4 shows the Bengali phone set.

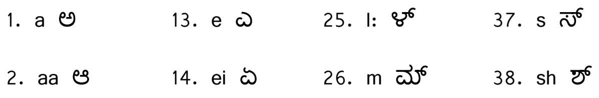


1. a অ	11. d:h ঢ্	21. jh ঝ্	31. n~ ঁ	41. t: ট্
2. aa আ	12. dh ধ্	22. k ক্	32. o ও	42. t:h ঠ্
3. ai ঐ	13. e এ	23. kh খ্	33. p ন্	43. t:k ৎ
4. au ঔ	14. g গ্	24. l ল্	34. ph ফ্	44. th থ্
5. b ব্	15. gh ঘ্	25. m ম্	35. r র্	45. u উ
6. bh ভ্	16. h হ্	26. n ন্	36. rx ঋ	46. uu ঊ
7. ch চ্	17. h: ঃ	27. n: ং	37. s স্	47. y য্
8. chh ছ্	18. i ই	28. nd~ ণ্	38. sh শ্	
9. d দ্	19. ii ঈ	29. ng~ ঙ্	39. shh ষ্	
10. d: ড্	20. j জ্	30. nj~ ঞ্	40. t ত্	

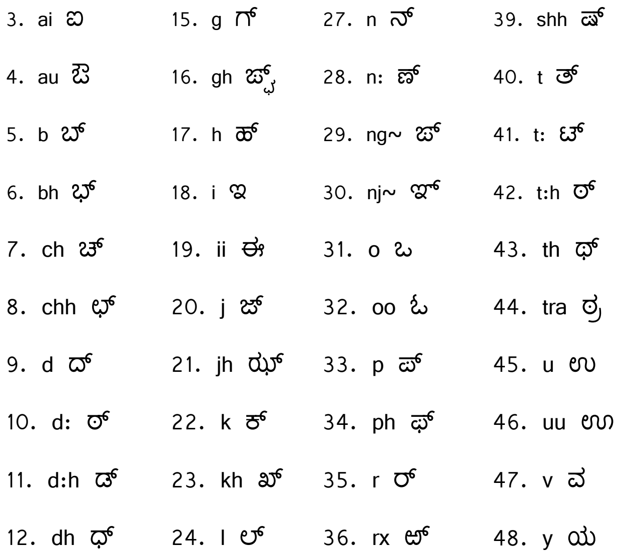
Figure 4: *47 Bengali phones*

3.2.3. *Kannada writing system*

The writing system of Kannada is the Telugu-Kannada script, which has origins in the Brahmi script and later developed into its current form [6]. There are twelve vowels in Kannada, of which eleven have both dependent and independent forms, and thirty four consonants. Each of the consonants also has an inherent vowel “a” just like Hindi which can be modified with the addition of dependent vowels. Fig. 5 shows the Kannada phone set.



1. a ಅ	13. e ಎ	25. l: ಳ್	37. s ಸ್
2. aa ಆ	14. ei ಏ	26. m ಮ್	38. sh ಶ್

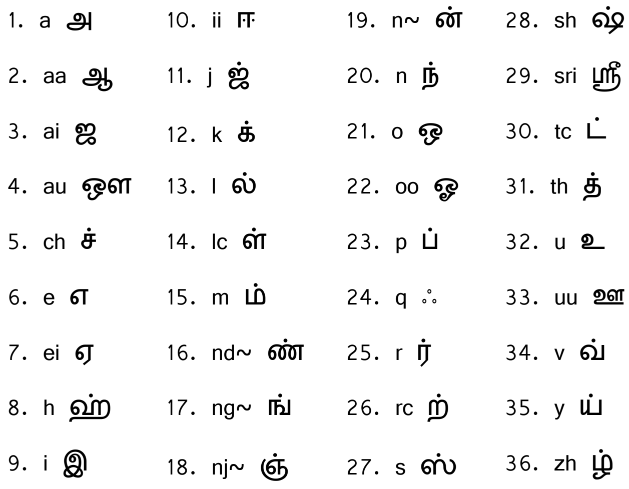


3. ai ಐ	15. g ಗ್	27. n ನ್	39. shh ಷ್
4. au ಔ	16. gh ಘ್	28. n: ಣ್	40. t ತ್
5. b ಬ್	17. h ಹ್	29. ng~ ಙ್	41. t: ಟ್
6. bh ಭ್	18. i ಇ	30. nj~ ಞ್	42. t:h ಠ್
7. ch ಚ್	19. ii ಈ	31. o ಒ	43. th ಥ್
8. chh ಛ್	20. j ಜ್	32. oo ಓ	44. tra ತ್ರ
9. d ದ್	21. jh ಝ್	33. p ಪ್	45. u ಉ
10. d: ಡ್	22. k ಕ್	34. ph ಫ್	46. uu ಊ
11. d:h ಢ್	23. kh ಖ್	35. r ರ್	47. v ವ
12. dh ಧ್	24. l ಲ್	36. rx ಋ್	48. y ಯ

Figure 5: *48 distinct Kannada monophones*

3.2.4. *Tamil writing system*

The writing system of Tamil is the Vatteluttu script [6]. There are twenty three consonants in Tamil, including a compound consonant (க்ஷ) of “k” and “sh”. Each consonant has an inherent vowel “a”, which can be changed to other vowels. The inherent vowel can also be silenced with diacritics, such as the Virama (்). There is a special consonant ஸ்ரீ which is never combined with any vowel [7]. Some of these consonants can also be modified by the Aytham (ஃ) to obtain sounds that are not present in Tamil [8]. However, these modified consonants were not present in the IIIT-H Indic Speech Database. Fig. 6 shows the thirty six (twelve vowels, twenty two simple consonants, Aytham and special consonant) phones of Tamil.



1. a அ	10. ii ஈ	19. n~ ன்	28. sh ஷ்
2. aa ஆ	11. j ஜ்	20. n ந்	29. sri ஸ்ரீ
3. ai ஐ	12. k க்	21. o ஒ	30. tc ட்
4. au ஔ	13. l ல்	22. oo ஓ	31. th த்
5. ch ச்	14. lc ள்	23. p ப்	32. u உ
6. e எ	15. m ம்	24. q ஃ	33. uu ஊ
7. ei ஏ	16. nd~ ண்	25. r ர்	34. v வ்
8. h ஹ்	17. ng~ ங்	26. rc ற்	35. y ய்
9. i இ	18. nj~ ஞ்	27. s ஸ்	36. zh ழ்

Figure 6: *36 distinct Tamil monophones*

3.2.5. *Punctuations*

For all the writing systems of the four languages, English punctuations are there. Punctuations commonly used include the full-stop (.), comma (,), exclamation mark (!), question mark (?) and colon (:). In addition to the English punctuation marks, the Danda (|) and double Danda (||) are used to mark the end of a sentence or verse [5]. The four languages are written from left to right and does not have distinct upper or lower cases. To form a word, letters are connected with a horizontal line on the top. A break in the ine represents the end of a word.

For Hindi, the Avagraha (ऽ) is used to show that a vowel is sustained in a cry or a shout [9]. Both Kannada and Tamil usually make use of English punctuation only [6]. In some specific cases, Devanagari punctuation as described above might be used.

3.3. G2P rules

In the above conversion between the Indian texts and phone sequence, mapping tables are used to map alphabets to their corresponding transliterations. This provides a simple way to synthesize the four languages without the use of dictionary. The following example illustrates this mapping. Same learning procedure is used for all the four languages.

The data files in IIIT-H Hindi collection consists of the Indian text and the corresponding phone-level transcription. This data pair is our reference for deriving the G2P rules. Given a phone set, the affiliated Hindi alphabets are deduced one by one until the alphabet-phone mapping converged. In Fig. 3, the resultant alphabet for each Hindi phone is shown.

To convert from the UTF-8 formatted Hindi texts to a phone sequence, the Indian texts are first read and their Unicode values are recorded. This sequence of Unicode values is then converted into a phone sequence by using the deduced alphabet-phone mapping. Fig. 7 depicts this process.

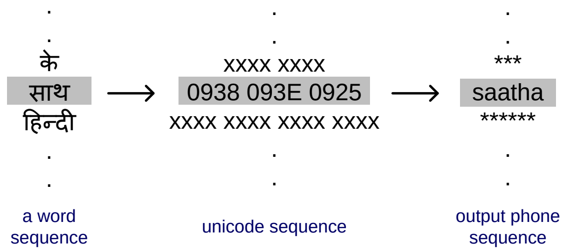


Figure 7: *The conversion process to obtain a phone sequence. xxx and *** here represent the Unicodes and phones*

3.4. System training and generation

Going through the speech data for the four languages, we found that the quality varies a lot. Some of the speech files are clean and highly intelligible; while some are suffered from severe background noise and/or reverberation. Hence, subsets of the speech data are selected based on the average segmental signal-to-noise ratio (segSNR). These subsets will be used for system training. Table 1 shows the details of the above data selection for each language.

language	measured segSNR range (dB)	min segSNR in selected subset (dB)	no. of sentences in selected subset
Hindi	[20.19, 37.43]	21	974
Bengali	[7.44, 35.71]	24	500
Kannada	[8.45, 35.5]	18	920
Tamil	[14.26, 39.81]	18	965

Table 1. *Details of selected data for training*

For a small amount of the speech files, the associated phone labels are found to be erroneous. These label files are revised before training.

Information about the spectrum, fundamental frequency (F0) and aperiodicity is embedded in our feature representation. STRAIGHT [10] is used for analysis and synthesis. Specifically, after STRAIGHT analysis, 40-th order mel-generalized cepstral (MGC) coefficients are extracted, together with the log energy, F0 and five-band aperiodicity values. These are the static features. Dynamic features, i.e. the delta and delta-delta features are also used. The resultant feature vector consists of 141 dimensions, split into five streams. One, three and one streams are used for spectrum, F0 and aperiodicity respectively. With these feature vectors, single-mixture, HMMs with diagonal covariance matrices are built. There are five left-to-right states in each HMM. The frame shift is 5 ms. Duration models are also made.

The training procedure is based on the steps shown in [4]. As speech files, text, phone labels and segment boundaries are merely available, the full-context label contains information about: the phone identity and position, phone number and position on word level, word number and position on sentence level.

During generation, the input text sentence is first converted to a full-context phone sequence with the above G2P mapping. Then, parameters for spectrum, F0 and aperiodicity are generated with the state sequence and models.

4. Evaluation Results

The following presents the major evaluation results for our submission to this year's Blizzard Challenge tasks. Our system is denoted as 'D'. System A refers to the real speech.

4.1. English

Based on the results of the listening evaluation conducted by the organizer, we summarize the performance of this year's engine in the following. The score of similarity is as shown in Fig. 8. From the figure, we can see that the similarity to the original speaker has a median value of 3. This indicates the generated voice retains the vocal characteristics of the speaker fairly well. The evaluation result of word error rate for SUS shows that the word error rate of our engine is at the middle range. This means the intelligibility of the voice is acceptable. From the MOS of naturalness, we realized that the naturalness of the voice is not as good as the similarity and intelligibility. This indicates the more effort needs to be put on naturalness in next year's evaluation. Especially, we need to put factors that accounts for prosody of passages into the processes of prosody prediction and voice synthesis.

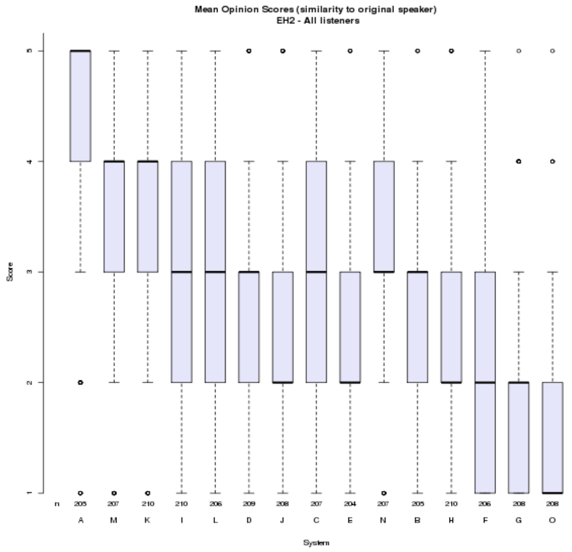


Figure 8: *MOS for similarity for English voice*

4.2. Indian Languages

The synthetic speech utterances were evaluated on three aspects: naturalness, similarity to the original speaker, and intelligibility (word error rate). Our system demonstrated similar performance over the four different languages. Figs. 9 and 10 show the mean opinion score distributions of our system on Hindi (IH1.1 task), WPD and SUS sentences respectively. These indicate that our generated speech was perceived to be unnatural.

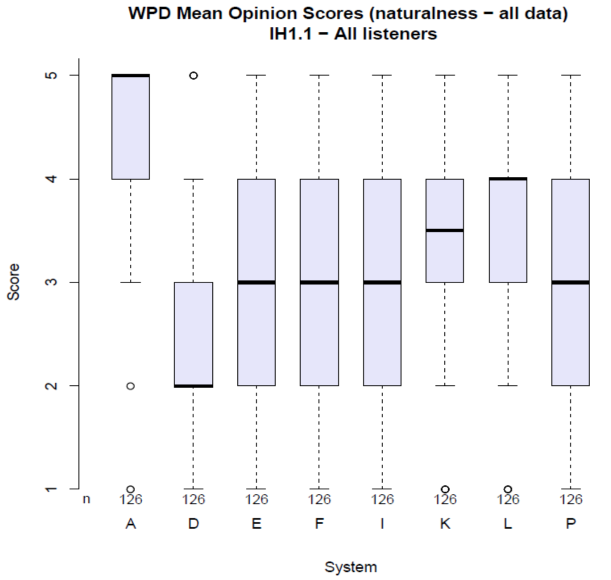


Figure 9: *MOS for naturalness on Hindi WPD sentences*

In the aspect of similarity to the original speaker, Fig. 11 (MOS for similarity on Hindi WPD sentences) shows that our generated speech was perceived to be dissimilar to the original speaker. This is probably due to the vocoder nature of the HMM-TTS architecture in our system.

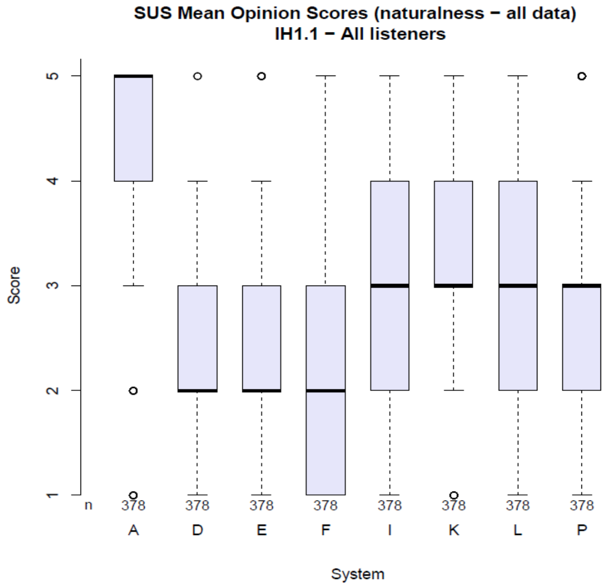


Figure 10: *MOS for naturalness on Hindi SUS sentences*

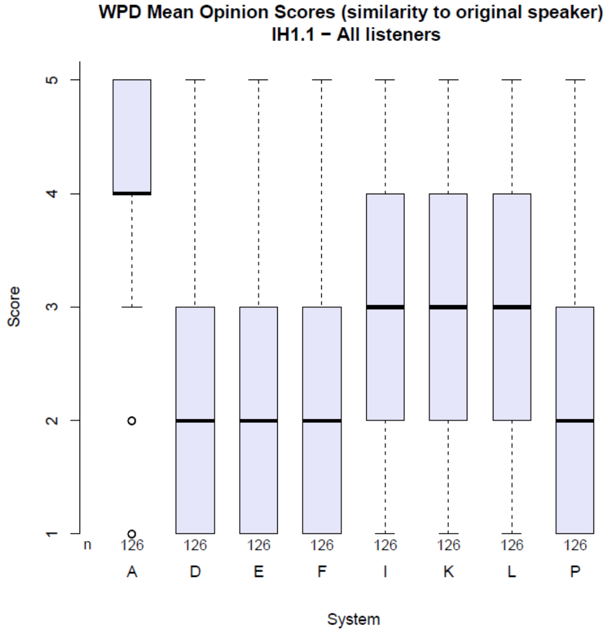


Figure 11: *MOS for similarity to original speaker on Hindi WPD sentences*

The strength of our system lies in the aspect of intelligibility. Figs. 12 and 13 are the word error rate (WER) plots for Hindi and Kannada SUS sentences. Note that WER was measured for these two languages only. Figs. 12 and 13 show that our HMM-TTS system achieved similar rates of intelligibility as real speech; the absolute differences in mean WER are 8% and 4% for Hindi and Kannada respectively.

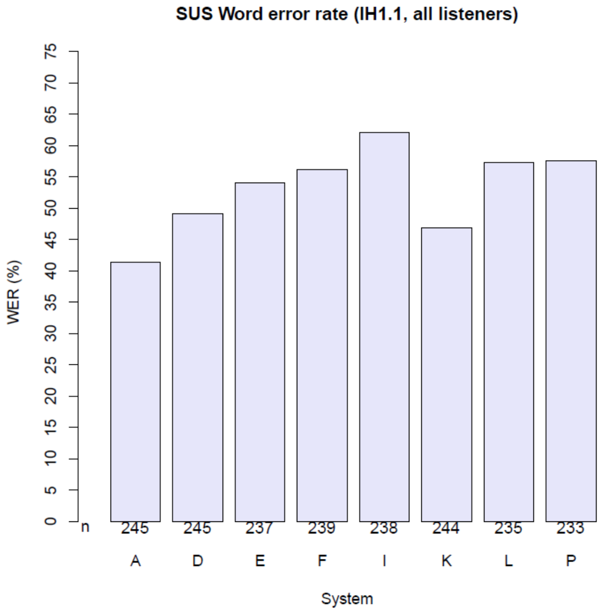


Figure 12: *WER on Hindi SUS sentences*

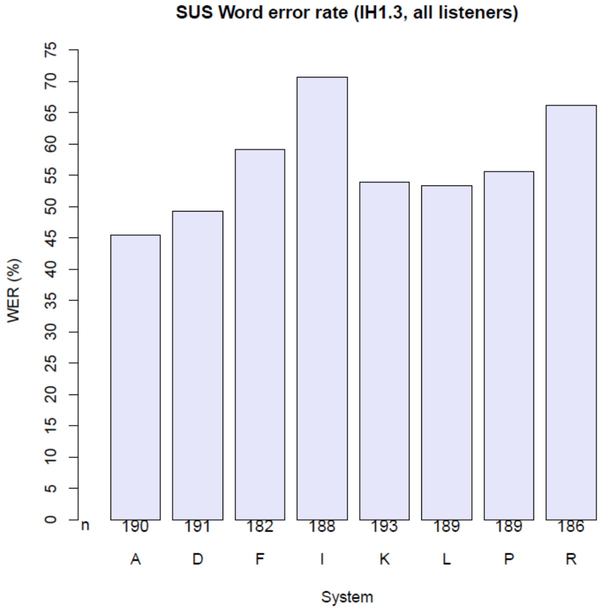


Figure 13: *WER on Kannada SUS sentences*

5. Conclusion

Much of the focus of Blizzard Challenge this year has been put on the new tasks on Indian languages. We participated in these Indian tasks, together with the usual English task as most of the year. With the considerations on the data sets provided, different synthesis systems are built. Unit-selection was used for the medium-sized English task (EH2); while HMM-based TTS was used for the small-sized Indian tasks (EH1.1 – EH1.4). With a typical HMM-TTS setup, our Indian voices have been found to achieve highly satisfactory intelligibility. Nevertheless, the evaluation on naturalness and the similarity are far from the expectation. Much effort is needed to build high-quality voices.

6. References

[1] K. Prahallad, E. N. Kumar, V. Keri, S. Rajendran, and A. W. Black, “The IIIT-H Indic speech databases”, *Proc. Interspeech*, Sep. 2012.

[2] The Unicode Consortium. The Unicode Standard, version 6.2.0 [Online]. Available: http://www.unicode.org/versions/ Unicode6.2.0/

[3] A. W. Black, H. Zen, and K. Tokuda, “Statistical parametric speech synthesis,” *Proc. ICASSP*, pp. 1229-1232, Apr. 2007.

[4] K. Tokuda, K. Oura, K. Hashimoto, S. Shiota, H. Zen, J. Yamagishi, T. Toda, T. Nose, S. Sako, and A. W. Black, (2011, Jul. 28) HMM-based speech synthesis system (HTS) [Online]. Available: http://hts.sp.nitech.ac.jp/

[5] R. Ishida, “An introduction to Indic scripts,” *Proc. of 23rd Internationalization & Unicode Conference*, Mar. 2003.

[6] B. Krishnamurti, “The Dravidian Languages”, Cambridge University Press, 2003.

[7] The Unicode Consortium. Tamil. [Online]. Available: http://www. Unicode.org/charts/PDF/U0B80.pdf

[8] E. Keane, “Tamil,” *Journal of the International Phonetic Association*, vol. 34, 2004, pp. 111–116.

[9] Wikipedia, (2013, Aug. 6) Devanagari [Online]. Available: http://en.wikipedia.org/wiki/Devanagari

[10] H. Kawahara, (2011, Jul. 28) STRAIGHT trial page. [Online]. Available: http://www.wakayama-u.ac.jp/STRAIGHTtrial/