Indian Language Screen Readers and Syllable Based Festival Text-to-Speech Synthesis System

Anila Susan Kurian, Badri Narayan, Nagarajan Madasamy, Ashwin Bellur,
Raghava Krishnan, Kasthuri G., Vinodh M.V., Hema A. Murthy
IIT-Madras, India
{anila,badri,nagarajan,ashwin,raghav,kasthuri,vinodh}@lantana.tenet.res.in
hema@cse.iitm.ac.in

Kishore Prahallad
IIIT-Hyderabad, India
kishore@iiit.ac.in

Abstract

This paper describes the integration of commonly used screen readers, namely, NVDA [NVDA 2011] and ORCA [ORCA 2011] with Text to Speech (TTS) systems for Indian languages. A participatory design approach was followed in the development of the integrated system to ensure that the expectations of visually challenged people are met. Given that India is a multilingual country (22 official languages), a uniform framework for an integrated text-to-speech synthesis systems with screen readers across six Indian languages are developed, which can be easily extended to other languages as well. Since Indian languages are syllable centred, syllable-based concatenative speech synthesizers are built.

This paper describes the development and evaluation of syllable-based Indian language Text-To-Speech (TTS) synthesis system (around festival TTS) with ORCA and NVDA, for Linux and Windows environments respectively. TTS systems for six Indian Languages, namely, Hindi, Tamil, Marathi, Bengali, Malayalam and Telugu were built. Usability studies of the screen readers were performed. The system usability was evaluated by a group of visually challenged people based on a questionnaire provided to them. And a Mean Opinion Score(MoS) of 62.27% was achieved.

1 Introduction

India is home to the world's largest number of visually challenged (VC) population [Chetna India 2010]. No longer do VC persons need to depend on others to access common information that others take for granted, such as newspapers, bank statements, and scholastic transcripts. Assistive technologies (AT), enable physically challenged persons to become part of the mainstream in the society. A screen reader is an assistive technology potentially useful to people who are visually challenged, visually impaired, illiterate or learning disabled, to use/access standard computer software, such as Word Processors, Spreadsheets, Email and the Internet.

Over the last three years, Indian Institute of Technology, Madras (IIT Madras) [Training for VC, IITM 2008], has been conducting a training programme for visually challenged people, to enable them to use the computer using a screen reader. The screen reader used was JAWS [JAWS 2011], with English as the language. Although, the VC persons have benefited from this programme, most of them felt that:

- The English accent was difficult to understand.
- Most students would have preferred a reader in their native language.
- They would prefer English spoken in Indian accent.
- The price for the individual purchase of JAWS was very high.

Although some Indian languages have been incorporated with screen readers like JAWS and NVDA, no concerted effort has been made to test the efficacy

Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pages 63–72,
Edinburgh, Scotland, UK, July 30, 2011. ©2011 Association for Computational Linguistics

of the screen readers. Some screen readers, read Indian languages using a non native phone set [acharya 2007]. The candidates were forced to learn by-heart the sounds and their correspondence to Indian languages. It has therefore been a dream for VC people to have screen readers that read using the native tongue using a keyboard of their choice.

Given this feedback and the large VC population ($\approx$ 15%) (amongst 6% physically challenged) in India, a consortium consisting of five institutions were formed to work on building TTS for six Indian languages namely Hindi, Telugu, Tamil, Bengali, Marathi and Malayalam. This led to the development of screen readers that support Indian languages, one that can be made freely available to the community.

This paper is organized as follows. Section 2 explains the selection of a speech engine, details of speech corpus, selection of screen readers and the typing tools for Indian languages. Section 3 discusses the integration of screen readers with Indian language festival TTS voices. Although the integration is quite easy, a number of issues had to be addressed to make the screen reader user-friendly. To do this, a *participatory design* [Participatory Design Conference 2011], approach was followed in the development of the system. Section 4 summarises the participation of the user community in the design of the system. To evaluate the TTS system, different tests over and above the conventional MOS [ITU-T Rec, P.85 1994], [ITU-T Rec, P.800 1996] were performed. Section 4 also describes different quality attributes that were used in the design of the tests. Section 5 provides the results of the System Usability Test. Section 6 provides details of the MOS evaluation conducted for the visually challenged community. Section 7 describes the future work and Section 8 concludes the paper.

2 Primary components in the proposed TTS framework

2.1 Selection of Speech Engine

One of the most widely used speech engine is *eSpeak* [espeak speech synthesis2011]. *eSpeak* uses ”formant synthesis” method, which allows many languages to be provided with a small footprint. The speech synthesized is intelligible, and provides quick responses, but lacks naturalness. As discussed in Section 1 the demand is for a high quality natural sounding TTS system.

We have used festival speech synthesis system developed at The Centre for Speech Technology Research, University of Edinburgh, which provides a framework for building speech synthesis systems and offers full text to speech support through a number of APIs [Festival 1998]. A large corpus based unit selection paradigm has been employed. This paradigm is known to produce [Kishore and Black 2003], [Rao et al. 2005] intelligible natural sounding speech output, but has a larger foot print.

2.2 Details of Speech Corpus

As part of the consortium project, we recorded a speech corpus of about 10 hours per language, which was used to develop TTS systems for the selected six Indian languages. The speech corpus was recorded in a noise free studio environment, rendered by a professional speaker. The sentences and words that were used for recording were optimized to achieve maximal syllable coverage. Table 1 shows the syllable coverage attained by the recorded speech corpus for different languages. The syllable level database units that will be used for concatenative synthesis, are stored in the form of indexed files, under the festival framework.

Language	Hours	No.Syll Covered
Malayalam	13	6543
Marathi	14	8136
Hindi	9	7963
Tamil	9	6807
Telugu	34	2770
Bengali	14	4374

Table 1: Syllable coverage for six languages.

2.3 Selection of Screen Readers

The role of a screen reader is to identify and interpret what is being displayed on the screen and transfer it to the speech engine for synthesis. JAWS is the most popular screen reader used worldwide for Microsoft Windows based systems. But the main drawback of this software is its high cost, approximately 1300 USD, whereas the average per capita income in India is 1045 USD [per capita Income of India 2011]

Different open source screen readers are freely available. We chose ORCA for Linux based systems and NVDA for Windows based systems. ORCA is a flexible screen reader that provides access to the graphical desktop via user-customizable combinations of speech, braille and magnification. ORCA supports the Festival GNOME speech synthesizer and comes bundled with popular Linux distibutions like Ubuntu and Fedora.

NVDA is a free screen reader which enables vision impaired people to access computers running Windows. NVDA is popular among the members of the AccessIndia community. AccessIndia is a mailing list which provides an opportunity for visually impaired computer users in India to exchange information as well as conduct discussions related to assistive technology and other accessibility issues [Access India 2011]. NVDA has already been intergated with Festival speech Engine by Olga Yakovleva [NVDA 2011]

2.4 Selection of typing tool for Indian Languages

The typing tools map the qwerty keyboard to Indian language characters. Widely used tools to input data in Indian languages are Smart Common Input Method(SCIM) [SCIM Input method 2009] and inbuilt InScript keyboard, for Linux and Windows systems respectively. Same has been used for our TTS systems, as well.

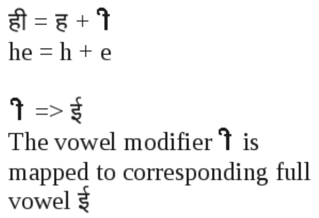
3 Integration of Festival TTS with Screen readers

ORCA and NVDA were integrated with six Indian language Festival TTS systems. Preliminary adaptations to the system for Indian languages are as follows.

- Though syllable based systems produce good quality speech output for syllabic Indian languages, syllables being larger units, require a large speech corpus to maximize syllable coverage. This means a larger footprint.

- In the paradigm being used, text processing modules are required to provide the syllable or phoneme sequence for the word to be synthesized. With input text for Indian languages being UTF-8 encoded, Indian language festival TTS systems have to be modified to accept UTF-8 input. A module was included in festival to parse the input text and give the appropriate syllable sequence. With grapheme to phoneme conversion being non-trivial, a set of grapheme to phoneme rules were included as part of the module.

- Indian languages have a special representation for vowel modifiers, which do not have a sound unit as opposed to that in Latin script languages. Hence, to deal with such characters while typing, they were mapped to sound units of their corresponding full vowels. An example in Hindi is shown in Figure 1.



ही = ह + ी
he = h + e

ी => ई
The vowel modifier ी is
mapped to corresponding full
vowel ई

Figure 1: Mapping of vowel modifiers

To enable the newly built voice to be listed in the list of festival voices under ORCA preferences menu, it has to be proclaimed as UTF-8 encoding in the lexicon scheme file of the voice [Nepali TTS 2008].

To integrate festival UTF-8 voice with NVDA, the existing driver, Festival synthDriver for NVDA by Olga Yakovleva was used [NVDA festival driver 2008]. To implement the rules for syllabifying Indian language text, a new C module was added to festival. Hence, festival [Festival compilation in Windows 2011] and synthDriver had to be recompiled [Compilation of NVDA Synthdriver 2011], for the new voice to be completely integrated with festival and usable under NVDA.

4 Participatory design

The design of the TTS system was arrived at, by active participation of visually challenged people, who

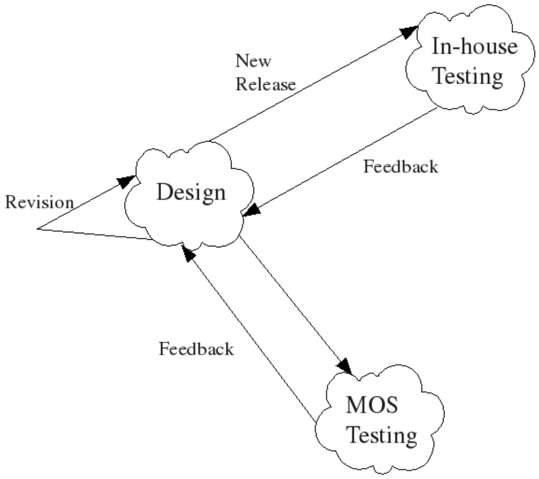


Figure 2: Flow of Development process

are the end users of the system. An educationally qualified visually challenged person was employed to test the integrated TTS system. The person is well versed in using JAWS screen reader on Windows. The quality attributes tested, were irrespective of languages. Hence, as a study, these tests were exclusively conducted on Tamil festival voice for both NVDA and ORCA.

When a new version of the system was released, it was provided to the in-house tester for evaluation. The suggestions and issues reported were then incorporated in the next release of the system. This process was done on an iterative basis. This helped in enhancing the system to meet the expectations of visually challenged people. Finally, the overall system performance was evaluated by conducting a Mean Opinion Score(MOS) test by visually challenged people, which is explained in detail in Sections 6. Figure 2 describes this development process.

The various quality attributes tested for, are :

- Usability of the TTS system
  - Adaptability of users to the new system
  - Navigation through desktop and web pages
- Availability of the TTS system
- Performance of the TTS system
  - Loading of voices
  - Response time for typing and reading

4.1 Usability of the TTS system

- **Adaptability of users to the new system**

  As the common screenreader used among the visually challenged community is JAWS, a study was conducted to find out the ease of adaptability for the user to the new system. Since the front end for the system are screen readers, the parameter used in this testing was primarily the learning involved in switching from JAWS to ORCA or NVDA. As JAWS and NVDA are Windows based screen readers, all the keystokes and shortcut keys are the same. A computer literate who has used JAWS, will learn NVDA quicker than others. As ORCA is a Linux based screen reader, the shortcut keys, key strokes and navigation through the system are different compared to that of JAWS. It takes more time for a JAWS user to familiarize with the Linux operating system, ORCA settings and keystokes.

- **Navigation of desktop and web pages using the screen reader**

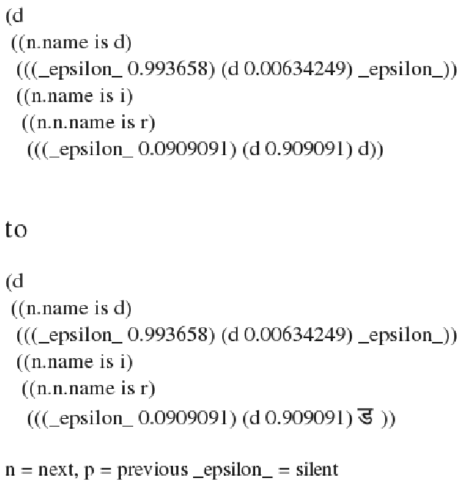
  When default *English locale* is selected for Windows and Linux systems, all the program menus and navigational keys are in English. The initial version of the TTS system was not able to parse these English words. As a solution, *switching of voices* between English and the selected Indian language was tried. The system was made to switch between Festival's English *Kal diphone* voice and one of the Indian language voices. When an English word is given as input, the English voice would be loaded and when an Indian language word is given as input, it switches to the respective Indian language, loads the voice and speaks the word. This frequent switching of voices degraded the performance of the system and hearing two different voices, without continuity was annoying to the listener. This led to the development of a bilingual voice.

  Bilingual Voice: Each Indian language voice is provided with an English pronunciation dictionary, so that when an English word is provided to the system, speech is synthesized using the Indian language voice itself. Following are the enhancements made to better the TTS system.

- **Pronunciation Dictionary for English words in native sound units**

  The English dictionary from Carnegie Mellon University(CMU) with phone pronunciation was used to create English to Native language pronounciation dictionary. An entry in the CMU dictionary :
  (”abolish” ax b aa l ih sh). These English phones were mapped to phones in native language. An example mapping from English to Hindi language :
  ax=अ , b=ब् , aa=आ, l=ल् , ih=इ , sh=श्.
  For all the English words in the dictionary, the native language representation was created, abolish = अबालिश्. The pronunciation dictionary was then created by breaking these words down into syllables and phone sequences present in the database.
  (”abolish” अ बा लिश्)
  All such English words that are required to navigate through a desktop(including special keys) and web, were collected and added to the pronunciation dictionary. The drawback of this method is that if an English word which is not present in the pronunciation dictionary, is provided as input, the TTS system cannot synthesize it. In order to overcome this, English Letter To Sound (LTS) rules were implemented.

- **Implementation of English LTS Rules**
  Inputs can be in English or the native language. In the case of a word being absent in the pronunciation dictionary, LTS rules should supplement. LTS rules have been developed for English in festival using a pronunciation dictionary of around 100000 words as the training set [Black et al. 1998]. These LTS rules generate a sequence of phones for the English word. By mapping the English phones to phones in the native language, one can provide a phone sequence in terms of the Indian language, for an English word. For example, a part of the Classification and Regression Tree(CART) for letter ’d’ in a word, by looking at the context in which it is occuring is shown in Figure 3. The first part of the figure is a partial tree in English. The second part of the figure is the corresponding entry for the Indian language. If ’d’ is followed by another ’d’, no sound(’epsilon;) is assigned. If it is followed by ’i’ and ’r’ ’phone d’ is assigned for English, whereas ’phone d’ is mapped to ड for Hindi language.



```
(d
 ((n.name is d)
  (((_epsilon_ 0.993658) (d 0.00634249) _epsilon_))
  ((n.name is i)
   ((n.n.name is r)
    (((_epsilon_ 0.0909091) (d 0.909091) d))


to

(d
 ((n.name is d)
  (((_epsilon_ 0.993658) (d 0.00634249) _epsilon_))
  ((n.name is i)
   ((n.n.name is r)
    (((_epsilon_ 0.0909091) (d 0.909091) ड ))

n = next, p = previous _epsilon_ = silent
```

Figure 3: CART for letter d

- **Recording of Common English words**
  Most of the English words when spoken in the Indian language voice did not sound intelligible enough. This is because, many English sounds were not available in Indian languages. Hence frequently seen English words while navigating a Windows/Linux desktop were recorded. Instead of concatenating Indian phones to synthesize the English word, the naturally uttered English word is spoken. This increased the usability of the system.

4.2 Availability of the TTS system

The system was tested to check if it responded to each and every input provided to it. Words, sentences and paragraphs were provided as input to the system using commonly used applications like notepad, word processor and browser. The system was able to read the words whose syllables were present in the database. The testing was done extensively for each language which resulted in some words not being spoken, which helped in the identification of those syllables which need to be included in the database. Some of the major issues identified during this test were:

- **Issues during typing**

  The evaluator tested the system by typing using SCIM in Linux systems and the inbuilt InScript keyboard in Windows systems. As it is unit selection based synthesis, the sound units for the corresponding characters that are picked up from the database may not be clear or audible. Also, the prosody of these individual characters, when taken from different contexts will vary. While typing, flat and prosodically neutral sounds are preffered. This led to recording of all *aksharas* (alphabets) in all six languages, in a prosodically neutral flat tone. It was also observed that the system was not reading vowel modifiers. This issue was solved by adding entries for vowel modifiers in the pronunciation dictionary. The vowel modifiers were mapped to the corresponding vowel pronunciation.

- **Issues during reading web pages**

  The system was tested for reading content from web pages. It was found that when a line with any special character(for example <,>,’) is given as input, the system would fail to read the entire line. This led to the handling of special characters in the Indian language voice. If anything outside the unicode range of the language is provided to the system, it is ignored. In this way, even if some special or junk characters are present in a line, the system will read the whole line ignoring these characters.

4.3 Performance of the TTS system

The evaluator noted the response time of the system while loading the voice, typing, navigation through desktop and web pages.

- **Loading of voices**

  In the unit selection paradigm, we have a large repository of multiple realizations of a unit (syllables) in different contexts. The text to be spoken is broken down into these units. Syllable speech units are then indexed with their linguistic and phonetic features using clustering techniques (CART) to capture the context in which they are uttered. With many realizations of the same syllable being present, CART are used to select a smaller set of candidate units for the syllable to be spoken. These CART built as part of the voice building process, attempt to predict the acoustic properties of the unit using its phonetic and linguistic features at the time of synthesis [Black and Taylor 1997].

  When the festival engine loads a voice, although the speech waveforms are saved on the hard disk, the CART gets loaded into the heap memory. As the size of this tree file exceeds the default heap size set in the festival framework, the initial version of the Indian language TTS voices failed to load. Hence, a larger heap size was provided as a runtime argument for the festival synthesizer.

- **Response time for typing and reading**

  The user expects the system to respond in a reasonable amount of time (approx 125 milliseconds). For the initial system, the response time for a sentence with 5 to 10 words was 1 to 2 seconds. To improve the response time of the system, the voice(s) had to be pruned. In the case of unit selection paradigm, a large database with multiple realizations is used to produce natural speech. Around 300000 units with multiple realizations of syllables including the ’silence’ unit are present in the database. In the cluster unit framework [Black and Taylor 1997], these syllables are clustered into similar sounding groups and form the leaves of the CART built. This resulted in a large CART file which in turn slowed down the system.

  With around 300000 realizations of syllables being present, it is seen that there are far too many realizations of frequently occuring syllables. So it was vital to prune the CART built. To effectively capture prosody for syllables, after experimenting heuristically with various cluster sizes, a leaf size of eight was used, i.e syllables are clustered into groups of eight. To prune the tree using the tools available within festival [Black and Lenzo 2000], within each cluster only two units closest to the centroid were retained and the rest were removed, hence reducing the tree size. Even after pruning the voice, it was seen that there were still a very large number (around 20000) of silence units, which are used to annotate phrase and sentence boundaries, in the speech corpus. It was seen that the silence units could be quantized into two units, one to denote end of phrase and another for end of sentence, without affecting the performance. Hence silence trees were removed from the CART retaining just the two quantized units, further pruning the tree and improving the speed. After pruning, the size of the tree for Tamil language was reduced from an 8 MB file to 1.7 MB file. The response time for sentences having word rate between 5 to 10 for the pruned system was 200milliseconds to 900 milliseconds. On an average there was 61% improvement in the response time.

5 System Usability Rating

For comparing the overall usability of the TTS system, before and after carrying out all the modifications listed in Section 4, a Usability test was conducted using screen readers by a group of visually challenged people. The System Usability Scale developed by John Brooke [Brooke 1996], which uses the Likert scale for providing a global view of subjective assessments of usability was used. The evaluators were provided with a questionnaire for which they have to provide Likert scale ratings. Table 2 shows the Likert scale used for the evaluation.

Scores	Scales
5	Strongly agree
4	Agree
3	Neither agree nor disagree
2	Disagree
1	Strongly disagree

Table 2: Likert Scales.

Questionnaire used for evaluation.

1. I found the system easy to use.
2. I need the support of a technical/non visually challenged person to be able to use the system.
3. I am able to navigate through Desktop and internet using the system without any help.
4. System is not able to clearly read each and every character I type.
5. Availability of the system is more than 90%. i.e. the system provides appropriate response to more than 90% of the input given to it.
6. Response time of the system is good and is within my tolerable limits.
7. I feel that most of the visually challenged people, having basic knowledge on computers, can learn this system quickly.
8. The system is not natural sounding.
9. The overall understanding/comprehensibility of the content read out by the system is high.
10. The system is very useful for the VC community.

The rating of the system was calculated as follows [Brooke 1996]. First, the score contributions from each item were summed up. Each item's score contribution will range from 0 to 4. The score contribution for positive questions 1,3,5,6,7,9 and 10 is the scale position minus 1. The score contribution for negative questions 2,4 and 8 is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of System Usability out of 100. A group of visually challenged people evaluated the initial and final system based on the questionnaire. The average System Usability score for the initial system was 35.63 and that of the final system was 89.38. Thus an improvement of around 50% in System Usability scores were seen due to the changes made in Section 4.

6 MOS Evaluation

MOS([ITU-T Rec, P.85 1994], [ITU-T Rec, P.800 1996]) and Degradation MOS (DMOS) tests were conducted for six Indian languages, across various centers in India. Synthesized speech files were played to the evaluators. Sentences belonging to different domains were chosen for quality evaluation, in order to test the performance of TTS system(s) upon receiving input text from varied domains.

The various factors that were considered, while administering the quality evaluation tests were:

- The MOS evaluators were chosen, such that they should not have participated in any listening quality test for synthetic speech, at least for the last 6 months and are well versed with the language.
- The tests were done up to a maximum of 30-40 minutes, in order to avoid listener fatigue.
- A reasonable number of MOS evaluators (a minimum of 20) were involved for evaluating the quality.
- The tests were conducted in a quiet room and the content to be evaluated was played through a good quality speaker.
- For MOS tests, the sentences belonging to various domains were grouped into various sets and the order of these sets were randomized in order to avoid any learning effects.
- Randomized sentences were played one after the other, with a brief pause for listeners to provide the quality score, based on the scales provided in Table 3.
- In the case of DMOS tests, a natural sentence followed by its synthesized counterpart is played after a brief pause and the listeners have to rate the amount of degradation in the synthesized sentence, relative to the natural sentence. This rating is based on the scales provided in Table 3.
- DMOS tests were conducted first, so that the participants get a feeling of how natural and synthesized sentences sound.

Figure 4: Active discussion among Visually Challenged candidates, during a training session

- 40 sentences were used to conduct the MOS test and 10 sentences for DMOS test.

The MOS and DMOS scores for the six Indian languages are provided in Table 4. Overall comprehension was also considered important, as the primary goal or aim of the TTS system was to be able to communicate information to the user. Thus, a preliminary comprehension based MOS test was conducted, which involved playing out a paragraph to the MOS evaluators and testing their level of comprehension.

Scores	Quality scales	
	MOS	**DMOS**
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Table 3: MOS and DMOS Scales.

7 Future Work

As a second phase of the project, we plan to carry out the following tasks

- To improve the prosody of synthetic speech.
- Enable the system to synthesize emotional speech.

Language	No. of Mos evaluators	News	Natural	Sports	InDomain	Science	DMOS	Overall MOS
Hindi	40	2.64	4.48	-	2.63	2.99	2.9	2.75
Bengali	8	3.31	-	2.91	3.18	2.85	3.14	3.06
Marathi	26	-	4.73	3.25	3.03	3.03	3.06	3.1
Telugu	23	-	4.66	2.46	2.89	2.83	3.68	2.73
Malayalam	27	3.95	4.13	3.73	3.77	-	3.91	3.82
Tamil	22	3.13	-	-	3.54	3.2	2.81	3.22

Table 4: Mos scores for six Indian languages.

- Build a small footprint TTS system, so that it can be used in applications for mobile, PDA, ATM etc.

- Evaluate the TTS system by conducting objective tests for intelligibility and naturalness, using different measures including the Semantically Unpredictable Sentence (SUS) test.

- To extend this effort to other Indian languages.

- To develop full-fledged Bilingual voices. In the current system we use the Indian language corpus to synthesize English words. The complete bilingual voice would have an English corpus recorded in the same voice as the Indian language, so that the same speech quality can be provided to both English and Indian language input.

8 Conclusion

In this paper, we have briefly discussed the efforts taken towards integrating TTS systems in six Indian languages, with screen readers ORCA and NVDA. We have also described the issues that were faced while testing the system and the solutions to improve the system. Further, results of the subjective listening tests (MOS and DMOS evaluation) and System Usability tests conducted were discussed.

With the completion of this project, training programme in IIT Madras, can be conducted for visually challenged community, using screen readers NVDA and ORCA for Indian Languages, instead of JAWS. Figure 4 shows an active discussion among visually challenged candidates during the computer training using screen readers at IIT Madras.

9 Acknowledgement

The authors would like to acknowledge the contributions of the Consortium members, namely IIIT hyderabad, IIT Kharagpur, CDAC Thiruvananthapuram and CDAC Mumbai, towards the project.

This project has been supported by the Department of Information Technology, India. (Project number - CSE0809107DITXHEMA).

References

S.P. Kishore and A.W. Black 2003. *Unit size in unit selection speech synthesis, proceedings of EUROSPEECH, pp. 1317-1320, 2003*

A.W. Black and P. Taylor 1997 *Automatically clustering similar units for unit selection in speech synthesis Eurospeech97 (Rhodes, Greece, 1997), vol. 2, pp. 601-604*

M. Nageshwara Rao,S. Thomas, T. Nagarajan and Hema A. Murthy 2005 *Text-to-speech synthesis using syllable like units, proceedings of National Conference on Communication (NCC) 2005, pp. 227-280, IIT Kharagpur, India, Jan 2005.*

ITU-T Rec, P.85 1997 *Method for Subjective Performance Assessment of the Quality of Speech Voice Output Devices, ITU-T Rec, P.85, 1994, Int. Telecom. Union*

ITU-T Rec, P.800 1996 *Methods for subjective determination of transmission quality, ITU-T Rec, P.800, 1996, Int. Telecom. Union*

A. Black and K. Lenzo 2000 *Building voices in the Festival speech synthesis system* `http://festvox.org/bsv/`

A. Black, P. Taylor, and R. Caley 1998 *The Festival speech synthesis system* `http://festvox.org/festival`

Festival Speech Synthesis System
`http://www.cstr.ed.ac.uk/projects/festival/`

ORCA Screen reader.
`http://live.gnome.org/Orca`

NVDA Screen reader.
`http://www.nvda-project.org/`

Festival synthDriver for NVDA by Olga Yakovleva:
`http://www.box.net/shared/jcnnzz7xu6`

SCIM Input method.
`http://apps.sourceforge.net/mediawiki/scim/index.php`
`https://help.Ubuntu.com/community/SCIM`

Festival compilation in Windows.
`http://www.eguidedog.net/doc_build_win_festival.php`

Participatory Design Conference
`http://www.publicsphereproject.org/drupal/node/235`

J. Brooke 1996 *“SUS: a ”quick and dirty” usability scale”. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, A. L. McClelland. Usability Evaluation in Industry.*
`http://hell.meiert.org/core/pdf/sus.pdf`

Nepali TTS User Manual. 2008.
`http://www.bhashasanchar.org/textspeech_intro.php`

JAWS Screen reader.
`http://www.freedomscientific.com/jaws-hq.asp`

espeak speech synthesis
`http://espeak.sourceforge.net/`

Training for Visually Challenged by IIT Madras
`http://www.lantana.tenet.res.in/TTSconsortiumWiki/doku.php?id=start/`

Acharya Multilingual Computing for literacy and education
`http://acharya.iitm.ac.in/` Last updated on 2007-03-19

Chetna India
`http://chetnaindia.org/our_values.htm`

Per capita Income of India, 2011
`http://www.financialexpress.com/news/Per-capita-income-in-India-is-Rs-46-492/744155/`

Access India mailing list
`http://accessindia.org.in/mailman/listinfo/accessindia_accessindia.org.in`

Compilation of NVDA Synthdriver
`http://www.lantana.tenet.res.in/TTSconsortiumWiki/doku.php?id=start`