Abstract— The Machine Translation domain of Natural Language Processing and the area of Artificial Intelligence are very useful in providing people with a machine, which understands diverse languages spoken by common people. It presents the user of a computer system with an interface, with which he feels more comfortable. Since, there are many different languages spoken in this world, we are constantly in need for translators to enable people speaking different languages to share ideas and communicate with one another. Human translators are rare to find and are inaccessible to the common man. With the concept of Machine Translation we may work towards achieving the goal using easily available computer systems. Besides, one of the first linguistic applications of computers to be funded was machine translation. The English to Marathi translator designed will be very useful to people who don’t understand English.

Speech synthesizer is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware.

Index Terms— Text to Speech, True Type Font, Natural Language Processing

INTRODUCTION

Currently Mobile industry is a growing market. Mobile devices are rather taking place of small computers with their improved processing speeds and memory. Not all devices have the same operating system. Some doesn’t even have an operating system that could support third party applications.

To resolve the problem of operating system multiple mobile manufacturing companies developed one solution. According to the solution all mobile devices implement common runtime environment called as J2ME, which supports third party applications made in J2ME.

However there are still limitations with the J2ME. One of them is to render True Type Font on device screen. The facility that we take for granted in computers is required to be implemented in J2ME. When certain graphical applications with a very good UI requires rendering fonts in other than system provided. This library provides a way to do it. It renders the fonts, which are vector based. Means they could be manipulated with. We can rotate them, zoom them, and move the text without having to worry about how they will look when we manipulate them. Rather it’s a precise rendering of it. This library allows the user to render the vector-based graphics to the mobile device screen which has following features. A simple True Type Font file is used to render the fonts, which is easily available.

1. Application based on this library allows user to translate and transmit the data in regular UTF-8 encoding which all the media transferring mediums support.
2. Hence all it takes it to render the same font again based on the UTF-8 characters.
3. One huge advantage is that programmer can localize their application regardless of whether support is available or not.

For example: MARATHI text based messenger based on TTF rendering library:

• If user has a mobile device, which doesn’t support Marathi fonts then using this application, user can write and read in Marathi language. However the scope of the project is not limited to one language. User can prefer to download other language fonts and store in mobile for rendering different language fonts.
• The user can set the text translation.
• Desktop based speech to text and translator.
• Developer can control over the text for how it is displayed on screen.
• Use of custom fonts is easily available with the help of this library.
• This library is highly reliable, precise and implements Java standards for implementing a library.

Language translation is a process in which a source language material is rewritten into a target communicates the same message, making it look like it has been originally written in that language[2]. Translation services play a vital role and getting an exact translation of your message is very important. The careful analysis of this module and its possible solutions lead to the following design of the system. The overall system design is diagrammatically shown below fig. 1.

Fig.1: Basic Block Diagram of Translator

Mrs.J.P. Kshirsagar, Prof. G. G. Chiddarwar
The text-to-speech (TTS) synthesis procedure consists of two main phases. The first one is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the acoustic output is produced from this phonetic and prosodic information. These two phases are usually called as high and low level synthesis.

The aim is to provide the current text to speech system with the ability to control speech parameters using a java language. The input text might be for example data from a word processor, standard ASCII from e-mail, a mobile text-message, or scanned text from a newspaper.

A simplified version of the procedure is presented in Figure 2. The character string is then preprocessed and analyzed into phonetic representation which is usually a string of phonemes with some additional information for correct intonation, duration, and stress [1]. Speech sound is finally generated with the low-level synthesizer by the information from high-level one. The simplest way to produce synthetic speech is to play long prerecorded samples of natural speech, such as single words or sentences. This concatenation method provides high quality and naturalness, but has a limited vocabulary and usually only one voice. The method is very suitable for some announcing and information systems. However, it is quite clear that we cannot create a database of all words and common names in the world. It may be even inappropriate to call this speech synthesis because it contains only recordings. Thus, for unrestricted speech synthesis (text-to-speech) we have to use shorter pieces of speech signal, such as syllables, phonemes, diphones or even shorter segments.

“Current text-to-speech systems are highly intelligible - indeed approaching the intelligibility of human speech. However, all systems sound unnatural, and the wealth of pragmatic information about the speaker and their state which is present in all human speech utterance is missing in present synthesis systems.” Therefore there is a real need for emotion in synthetic speech and it would be a valuable addition to this synthesizer to be able to speak with emotion [1]. Current Text to speech does not use emotion or prosody except poorly via Microsoft’s Speech API. In addition to this, text to speech systems are an important part of virtual conversational characters. With the use of a text to speech system it is possible to divulge real-time, personalized content given that text to speech systems are not limited by their vocabulary in the same way as systems that rely on prerecorded sentences.

A. Translation

The correct translation of the sentence largely depends on the parse that is obtained for it. So, depending on the language in which it has to be translated, the details of the parse of each word should be designed appropriately.

B. Generation

Once the parse gives the details of each word, the corresponding phrase in English needs to be generated. An important aspect of translation is the grammatical structure of the sentences. Different languages follow varied syntactical structures. While Marathi follows a free ordering of sentences, English sentences follow SVO (Subject Verb-phrase Object) format for active and OVS (Object Verb-phrase Subject) format for passive sentences.

C. Ambiguities

Each individual word in Marathi corresponds to a word or a phrase in English. Also, there are many syntactically correct equivalent phrases of English corresponding to each word of Marathi. Resolving this ambiguity needs extra knowledge categorizing words of the sentence according to their peculiarities. When we deal with nouns of Marathi, corresponding phrases of English are nouns of English along with appropriate prepositions. The prepositions accompanying each noun depend on the case. For each case in Marathi, there is a set of possible prepositions. Resolving this ambiguity needs a through study of individual words and word categories in general. But we know from experience that it is more reasonable to consider the probability of occurrence of something done ‘by’ the stick than something being obtained ‘from’ the stick.

So, three approaches to solve this problem would be - to consider the preposition with higher probability of occurrence for all nouns of Marathi, or to store knowledge of the appropriate preposition for each noun, or to determine the most suitable preposition from a combination of the word knowledge and the semantics of the sentence.

II. PERFORMANCE ANALYSIS FOR SYNTHESIZER
The major feature of a speech synthesizer that affects its understandability, its acceptance by users and its usefulness to application developers is its output quality. Knowing how to evaluate speech synthesis quality and knowing the factors that influence the output quality are important in the deployment of speech synthesis.

Humans are conditioned by a lifetime of listening and speaking. The human ear (and brain) is very sensitive to small changes in speech quality. A listener can detect changes that might indicate a user’s emotional state, an accent, a speech problem or many other factors. The quality of current speech synthesis remains below that of human speech[1], so listeners must make more effort than normal to understand synthesized speech and must ignore errors. For new users, listening to a speech synthesizer for extended periods can be tiring and unsatisfactory.

The two key factors a developer must consider when assessing the quality of a speech synthesizer are its understandability and its naturalness. Understandability is an indication of how reliably a listener will understand the words and sentences spoken by the synthesizer. Naturalness is an indication of the extent to which the synthesizer sounds like a human - a characteristic that is desirable for most, but not all, applications. Understandability is affected by the ability of a speech synthesizer to perform all the processing steps described above because any error by the synthesizer has the potential to mislead a listener. Naturalness is affected more by the later stages of processing, particularly the processing of prosody and the generation of the speech waveform. Though it might seem counter-intuitive, it is possible to have an artificial-sounding voice that is highly understandable. Similarly, it is possible to have a voice that sounds natural but is not always easy to understand (though this is less common).

A. Factors Affecting Overall Quality of a TTS system

When we create a text-to-speech (TTS) solution, our choice of a TTS system is naturally critical. Below are the main contributing factors that we believe affect the overall quality of a TTS system.

- Intelligibility – How much of the spoken output can we understand? How quickly do we become fatigued as a listener?
- Naturalness – How much like real speech does the output of the TTS system sound?
- Front-end processing – The ability of the system to deal intelligently with commonly used challenges in text such as abbreviations, numerical sequences, homographs and so on.

Having defined these three categories, it’s clear that they are not independent of one another; serious front-end errors will lead to less intelligible speech, and this will be perceived as less natural. Although no evaluation will give you a full proof answer, a thorough assessment should provide us with useful and relatively reliable data to help select a TTS system.

III. EXISTING CHALLENGES

Before discussing the proposed TTS system, we need to address many research challenges related to TTS. Specifically following are few critical ones which need to be concentrates on Text normalization, Text-to-phoneme challenges and evaluation.

A. Text normalization challenges

The process of normalizing text is rarely straightforward[3]. Texts are full of heteronyms, numbers, and abbreviations that all require expansion into a phonetic representation. There are many spellings in English which are pronounced differently based on context. For example, “My latest project is to learn how to better project my voice” contains two pronunciations of “project”.

B. Text-to-phoneme challenges

Speech synthesis systems use two basic approaches to determine the pronunciation of a word based on its spelling, a process which is often called text-to-phoneme or grapheme-to-phoneme conversion (phoneme is the term used by linguists to describe distinctive sounds in a language). The simplest approach to text-to-phoneme conversion is the dictionary-based approach, where a large dictionary containing all the words of a language and their correct pronunciations is stored by the program. Determining the correct pronunciation of each word is a matter of looking up each word in the dictionary and replacing the spelling with the pronunciation specified in the dictionary. The other approach is rule-based, in which pronunciation rules are applied to words to determine their pronunciations based on their spellings. This is similar to the "sounding out", or synthetic phonics, approach to learning reading. Each approach has advantages and drawbacks. The dictionary-based approach is quick and accurate, but completely fails if it is given a word which is not in its dictionary. As dictionary size grows, so too does the memory space requirements of the synthesis system. On the other hand, the rule-based approach works on any input, but the complexity of the rules grows substantially as the system takes into account irregular spellings or pronunciations. (Consider that the word "of" is very common in English, yet is the only word in which the letter "f" is pronounced.) As a result, nearly all speech synthesis systems use a combination of these approaches.

C. Evaluation challenges

It is very difficult to evaluate speech synthesis systems consistently because there is no subjective criterion and usually different organizations use different speech data. The quality of a speech synthesis system highly depends on the quality of recording. Therefore, evaluating speech synthesis systems is almost the same as evaluating the recording skills.

Recently researchers start evaluating speech synthesis systems using the common speech dataset. This may help people to compare the difference between technologies rather than recordings.

IV. CONCLUSION

The concept is to build the proposed methodology is purely research oriented. Speech synthesis has been developed steadily.
over the last decades and it has been incorporated into several new applications. For most applications, the intelligibility and comprehensibility of synthetic speech have reached the acceptable level. However, in prosodic, text preprocessing, and pronunciation fields there is still much work and improvements to be done to achieve more natural sounding speech. With good modularity it is possible to divide the system into several individual modules whose developing process can be done separately if the communication between the modules is made carefully.

The basic methods used in speech synthesis have been introduced in the most commonly used techniques in present systems are based on formant and concatenative synthesis. The latter one is becoming more and more popular since the methods to minimize the problems with the discontinuity effects in concatenation points are becoming more effective. The concatenative method provides more natural and individual sounding speech, but the quality with some consonants may vary considerably and the controlling of pitch and duration may be in some cases difficult, especially with longer units. With concatenation methods the collecting and labeling of speech samples have usually been difficult and very time-consuming.

TTS in Java is a new area of research and these synthesizer benefits of all the advantages of the language. Therefore it is important to provide it with equivalent functionality to other TTS systems in order for these advantages to be of any use. Clearly, speakIt - TTS systems offer many significant advantages over prerecorded audio, including
1. Size,
2. Vocabulary
3. Customizability
4. Cost
5. Flexibility

Furthermore, in terms of voice quality, TTS systems produce very intelligible speech output. Formant TTS voices are typically not as natural-sounding as concatenative TTS voices, but both provide capabilities that prerecorded audio cannot, most notably, the ability to present unbounded, dynamic information to the user.

REFERENCES


First Author Mrs. J.P.Kshirsagar pursuing M.E.(Comp.Net.) from Sinhgad College of engineering. Since 2006 working in JSPM. Presently working as lecturer in Department of Computer Engineering, Rajarshi shahu college of engineering,Tathawade, Pune-411033. Affiliated to University of Pune, Maharashtra, India. ISTE Life Membership and published 2 journal papers and 3 international papers.Area of interest is NLP.

Second Author Prof. G. G. Chiddarwar received M.E. in Computer from Pune institute of Computer Technology, University of Pune in May 2005. Since 2000 working in Sinhgad Technical Education Society. Presently working as Assistant Professor in Department of Computer Engineering, Sinhgad College of Engineering, Affiliated to University of Pune, Maharashtra, India. ISTE life membership and published three journal papers and three international conference papers. Areas of interest are NLP, Image Processing, and Network Security.