A simple phoneme based speech recognition system

Sadhana Gopal, Trishant Malik, Seema Devi

***Abstract*— To build a natural sounding synthesis system, it is essential that the text processing component produce an appropriate sequence of phonemic units corresponding to an arbitrary input text. This paper describes a text-to-speech system (or "engine"), composed of two parts: a front-end and back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end – often referred to as the synthesizer- then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.**

***Index terms*- front-end processing, normalization, phonemes, speech output, synthesizer**

I. INTRODUCTION

Speech synthesis enables voice output by machines or devices. Text-to-speech synthesis does so by using text as input. The goal of text-to-speech system synthesis is to convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person. Therefore, the system goes beyond simple 'cut and paste' system used, for example, in some telecom applications to read back a phone number. Such systems string together words spoken in isolation and the artifacts of such a scheme are often perceptible.

The methodology used in this system is to exploit acoustic representations of speech for synthesis, together with linguistic analyses of text to extract correct pronunciations and prosody in context. Synthesis systems are commonly evaluated in terms of three characteristics: accuracy of rendering the input text, intelligibility of the resulting voice message, and perceived naturalness of the resulting speech.

We distinguish a system's front-end (the part of the system close to the text input) from the system's back-end (part of the system closer to speech output). Input text, optionally enriched by that control prosody or other characteristics, enters the front-end where a text analysis module detects the document structure, followed by text normalization (expansion to literal word tokens, encompassing transcription of acronyms, abbreviations, currency, dates, times, URLs, etc..) and further linguistic analysis that enables other tasks down the line. The tagged text then enters a phonetic analysis block that performs homograph disambiguation, and grapheme to phoneme conversions. The string of tagged phones enters into the prosodic analysis block that determines pitch, duration (and amplitude) targets for each phones. Finally, the string of symbols that was derived from a given input word or text is passed on to the speech synthesis module, where it controls the voice rendering that corresponds to the input text.

We have attempted to create a system that generates sound with every key the user inputs. The sound of the phoneme is generated as the user presses a key. The sounds of the phonemes are stored in templates, which are matched while processing the text input corresponding to a key. Since the system generates sound as and then, the system can be used to generate sound or pronounce any word in any language.

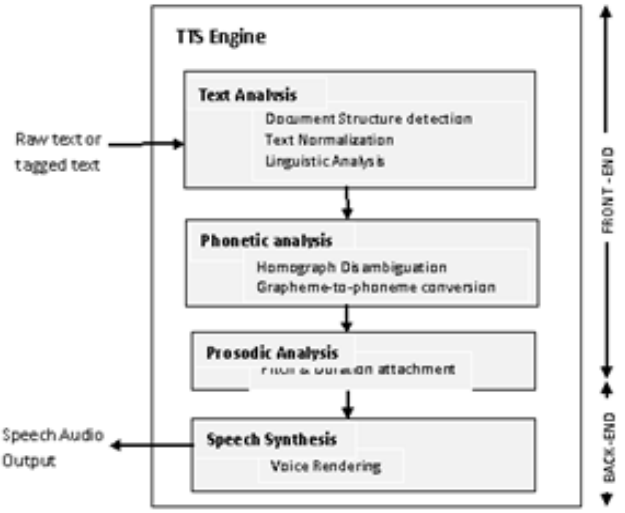

Fig.1: Block diagram of a concatenative text-to-speech system showing the front-end and back-end stages

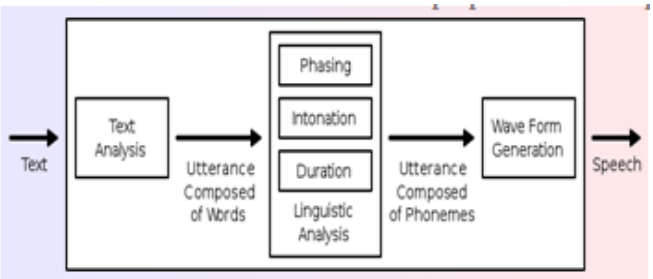

Fig. 2: General Block Diagram of a text-to-speech system

Manuscript received April 15, 2014

Sadhana Gopal, Computer Science, Maharshi Dayanad Universsity,Dronacharya College of Engineering, Gurgaon, India, 9560086374.

Trishant Malik, Computer Science, Maharshi Dayanad Universsity,Dronacharya College of Engineering, Gurgaon, India, 9811904225.

Seema Devi, Computer Science, Maharshi Dayanad Universsity,Dronacharya College of Engineering, Gurgaon, India, 9654795967.

II. PREPROCESSING PHASE

The text analysis and normalization module in the front-end determines to a large extend the 'what' and 'how' of the resulting synthetic speech. The punctuations are not infallible. For example, the system should not misinterpret the dot after 'in' in the example, "The box is 40.5 in. long" as the end of the sentence. In addition, punctuation and other special characters can be a part of a time, date, or currency expression. Text normalization is difficult because it is context sensitive (e.g., $2.6 million = two point six million dollars).

Abbreviations and acronyms fall in either of two categories. The first category contains a finite set of known 'mappings' such as "Dr." in the sentence "Dr. William lives on Smith Dr.". Note that a mapping may be ambiguous. In this case, "Dr." can refer to either 'doctor' or 'drive'. More difficult to handle, however, is the open category of abbreviations that people invent on the fly. Therefore, it may be necessary to handle certain text. Normalization tasks in form of a domain specific text "filter" that would alter the raw text before it passed on to the system as depicted in the figure. Applications like e-mail reading or web page reading, for example, also requires text filters to strip out mundane header or formatting information. Even the simple reading of number can be difficult, such as '370', where the 370 can be part of a phone number (370- 11 11, read as "three-seven-zero. . .") or part of a name (e.g., IBM370, read as "i-b-m-three-seventy"). Thus, the performance of the text analysis and normalization module affects the accuracy rating of a text-to-speech system. Linguistic analysis in the front-end encompasses the determination of parts of speech (POS), word sense, emphasis, appropriate speaking style, and speech acts. A linguistic parser could be used, but typically only a shallow analysis is done for computational speed.

III. SYNTHESIZER TECHNOLOGIES

The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

The two primary technologies generating synthetic speech waveforms are concatenative synthesis and formant synthesis. Each technology has strengths and weaknesses, and the intended uses of a synthesis system will typically determine which approach is used.

A. Concatenative synthesis

Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. Generally, concatenative synthesis produces the most natural-sounding synthesized speech. However differences between natural variations in speech and the nature of the automated techniques for segmenting the waveforms sometimes result in audible glitches in the output.

B. Formant synthesis

Formant synthesis does not use human speech samples at runtime. Instead, the synthesized speech output is created using additive synthesis and an acoustic model (physical modeling synthesis). Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called rules-based synthesis; however, many concatenative systems also have rules-based components. Many systems based on formant synthesis technology generate artificial, robotic-sounding speech that would never be mistaken for human speech. However, maximum naturalness is not always the goal of a speech synthesis system, and formant synthesis systems have advantages over concatenative systems. Formant –synthesized speech can be reliably intelligible, even at very high speeds, avoiding the acoustic glitches that commonly plague concatenative systems.

Text normalization challenges

The process of normalizing text is rarely straightforward. 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".

Recently TTS systems have begun to use HMMs to generate "parts of speech" to aid in disambiguating homographs. This technique is quite successful for many cases such as whether "read" should be pronounced as "red" implying past tense, or as "reed" implying present tense. Typical error rates when using HMMs in this fashion are usually below five percent. These techniques also work well for most European languages, although access to required training corpora is frequently difficult in these languages.

Deciding how to convert numbers is another problem that TTS systems have to address. It is a simple programming challenge to convert a number into words (at least in English), like "1325" becoming "one thousand three hundred and twenty- five." However, numbers occur in many different contexts; "1325" may also be read as "one three two five", "thirteen twenty-five" or "thirteen hundred and twenty five". A TTS system can often infer how to expand a number based on surrounding words, numbers, and punctuation, and sometimes the system provides a way to specify the context if it is ambiguous. Roman numerals can also be read differently depending on context. For example "Henry VIII" reads as "Henry the Eighth", while "Chapter VIII" reads as "Chapter Eight".

Similarly, abbreviations can be ambiguous. For example, the abbreviation "in" for "inches" must be differentiated from the word " in", and the address "12 St John St." uses the same abbreviation for both "Saint" and "Street". TTS systems with intelligent front-ends can make educated guesses about ambiguous abbreviations , while others provide the same result in all cases, resulting in nonsensical (and sometimes comical) outputs, such as "co-operation" being rendered as "company operation".

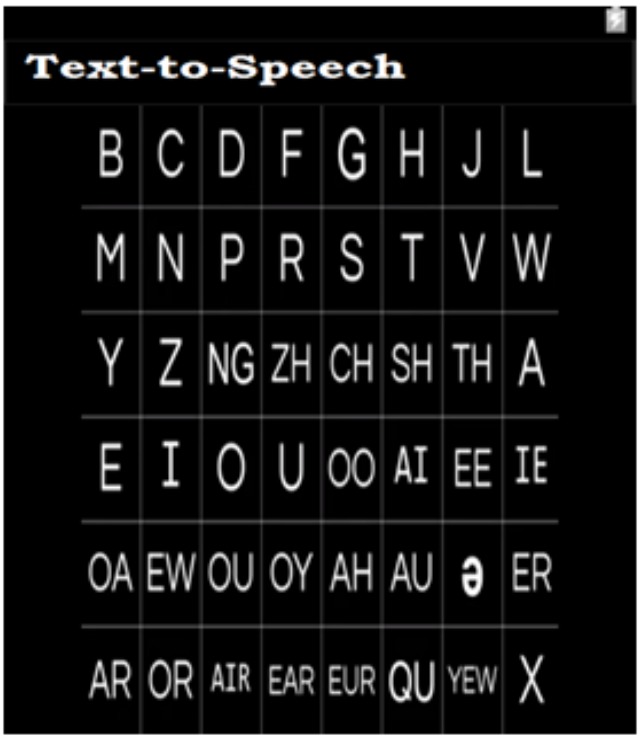

Fig.3: Screen-shot of the implementation

IV. 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.

V. Applications

Speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread. It allows environmental barriers to be removed for people with a wide range of disabilities. The longest application has been in the use of screen readers for people with visual impairment, but text-to-speech systems are now commonly used by people with dyslexia and other reading difficulties as well as by pre-literate children. They are also frequently employed to aid those with severe speech impairment usually through a dedicated voice output communication aid.

Speech synthesis techniques are also used in entertainment productions such as games and animations.

In recent years, Text to Speech for disability and handicapped communication aids have become widely deployed in Mass Transit. Text to Speech is also finding new applications outside the disability market. For example, speech synthesis, combined with speech recognition, allows for interaction with mobile devices via natural language processing interfaces.

VI. Conclusion

We would conclude by highlighting some aspects of Text-to-Speech (TTS) synthesis with a slant towards catering to electrical engineers. Many aspects, such as, for example, prosody generation, natural language processing, and others, have been skimmed only for space reasons. It is clear that TTS systems have come a long way towards delivering high-quality output to users that sometimes fools them to believe that they are listening to recordings. This said, it is also clear that we are still far from delivering the perfect synthesis for all possible applications. One of the most important finding of this paper is that we achieved very good phoneme recognition accuracy with a very simple phoneme model. Despite the fact that the phoneme recognition accuracy was not increased, our simple phoneme recognition system warrants stable and reliable behavior with a good recognition and synthesis performance.

References

[1] C. Bickley, A. Syndral, and J. Schroter, "Speech Synthesis", in The Acoustics of Speech Communication.
[2] T. Dutoit, An Introducion to Text-to-Speech synthesis.
[3] M.M.Sondhi and D.J.Sinder, "Arculatory Modeling: a role in concatenative text-to-speech synthesis" in Text-to-Speech Synthesis: New Paradigms and advances.
[4] Lee, K-F., Hon, W-F., Speaker independent phone recognition using hidden Markow Models, IEEE Transactions on Acoustics, Speech and Signal processing.
[5] J.P.H. Van Santen, "Combinatorial Issues in text to speech synthesis".
[6] M.J. Makashay, C.W. Whiteman, A.K.Sydral and A.D.Conkie, "Perceptual Evaluation of automatic segmentation in text-to-speec synthesis".
[7] Y.Yagiska, N.Kaiki, N.Iwahashi, and K.Mimura, "ATR-v-Talk speech synthesis system".
[8] O. Fujimura and J.Lovins, "Syllables as concatenative phonetic elements".
[9] W. Klejin and K. Paliwal, Eds, Speech Coding andSynthesis.