International Journal of Soft Computing 3 (3): 224-229, 2008
ISSN: 1816-9503
© Medwell Journals, 2008

A Text to Speech Interface for a Digital Library

N. Puviarasan, P. Aruna and S. Palanivel
Department of Computer Science and Engineering, Faculty of Engineering and Technology,
University of Annamalai, Annamalainagar, Chidambaram-608 002, India

Abstract: This study focuses on design and implementation of Text to Speech (TTS) system, for converting English text into speech signal. It generates speech from the phonetic transcripts of text. It comprises of digitization and text-to-speech generation. The objective of Text to Speech Interface for a Digital Library (TT SIDL) is to store the information of the book in the digital format. Then, a text to speech system would enable to access the digital content of text and produce its corresponding voice. It helps to illiterate and vision-impaired people, for hearing and quick understanding of the contents of the book. This system involves various processes like text-normalization subsuming sentence segmentation and normalization of non-standard words, statistical Part-Of-Speech tagging (POS), grapheme-to-phoneme conversion and prosodic analysis. Subsequently, the system allows the user to select the required text and generates the speech, which has good quality in naturalness and intelligibility.

Key words: TTS, TTSIDL, OCR, POS, grapheme-to-phoneme conversion, normalization of non-standard words

INTRODUCTION

The process of a text-to-speech system for a digital library starts with the process of digitization. The digitization begins with the page by page scanning of the book. Then, the digitized book is in the form of series of images where each image corresponds to a page in the book. Each digitized page is processed using optical character recognition to obtain the text. This digitized text is stored in the digital library portal. Based on the request from the user, the text is sent to a text to speech system for converting into speech signal.

The goal of text to speech system is to convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to person. Most of the digital information present in the digital world is accessible to a few who can read or understand a particular language. Language technologies can provide solutions in the form of natural interfaces so that digital content can reach the masses and facilitate the exchange information across different people.

Speech 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 resulting speech. Today, applications of TTS are in automated telecom services, as a part of a network voice server for e-mail, in directory assistance, as an aid in providing up-to-the-minute information to a telephone user, in computer games and last but not least, in aids to the handicapped. The scope of this work is to design and implement a Text To Speech (TTS) interface for a digital library portal primarily for an English language. This study address the issues involved in building interface and demonstrate the text to speech system for English language.

PREVIOUS WORKS ON TEXT-TO-SPEECH SYSTEM

In this study, the research works related to text-to-speech is explained.

The IBM Mandarin Text-To-Speech (TTS) system Jiang *et al*. (2006) is a concatenative synthesis system. Synthesis units are selected from a large corpus based on the prosody model and other context features. In most Mandarin speech synthesis system, syllable is used as the basic synthesis unit. Then, the selected syllables are concatenated together and the waveform is generated. Pitch and duration features of the segments are usually not modified to preserve the acoustic quality, except some special requirements are encountered. For example, to provide a web-page reader to blinds, the duration is modified to shorter time to make the speech faster and more information can be given during the same period of time. The system is the start-of-the-art with high quality. By using data-driven methods, the system also has the capability to develop new voices fastly.

Corresponding Author: N. Puviarasan, Department of Computer Science and Engineering, Faculty of Engineering and Technology, University of Annamalai, Annamalainagar, Chidambaram-608 002, India

Incorporation of speech and Indian scripts can greatly enhance the accessibility of web information among common people. Indian accent text-to-speech system (Aniruddha and Samudravijaya, 2002) for web browsing describes a 'web reader', which 'reads out' the textual contents of a selected web page in Hindi or in English with Indian accent. The content of the page is downloaded and parsed into suitable textual form. It is then passed on to an indigenously developed text-to-speech system for Hindi/Indian English, to generate spoken output. The web reader detects the hypertext links in the web pages. It gives the option to the user to follow the link or continue scrutinize the current web page. Future plans include refining the web parser, improvement of naturalness of synthetic speech and improving the robustness of the speech recognition system.

The next development in this field is a papageno TTS system (Hain *et al*., 2006) for embedded systems like mobile phones or PDAs. Restrictions like CPU, bandwidth and memory consumption have to be considered during the design of the system. Several steps are taken to save memory, for instance the weights of the neural networks are stored as 8bit fixed-point values instead of 32bit floating-point values. Another reason to use fixed-point arithmetic is that most CPUs in embedded systems do not have a floating-point unit. The speaker database is compressed and only small lexica are used. The goal for the development within the TC-STAR project is a high-quality TTS system for UK English. In the text preprocessing, more detailed rules or algorithms are applied, bigger neural networks and huge lexica are used. The prosody generation is based on more information and also uses bigger neural networks.

FUNCTIONAL PROCESS OF TTSIDL

The TTSIDL consists of the following four functional modules. The block diagram representation of TTSIDL is shown in Fig. 1. This study depicts the functional process.

- Digitization of book
- Optical Character Recognition (OCR) Process
- Digital Library Portal Formation
- Text-To-Speech System

The process of digitization begins with the page by page scanning of the book. So, digitized book is a series of images where each image corresponds to a page in the book. Each digitized page is processed using Optical Character Recognition (Susan, 1996) to obtain text in ASCII or Unicode format. The digitized text is stored in the digital library portal. Based on request from the user, the selected text is sent to a text to speech system for conversion into a speech signal.

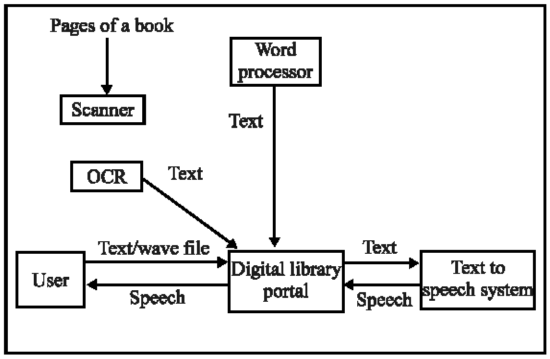


Fig. 1: Block diagram representation

Digitization: In the library context, digitization usually refers to the process of converting a paper or film-based document into electronic form. The electronic conversion is accomplished through imaging a process whereby a document is scanned and an electronic representation of the original is produced in the form of a bitmap image. Optical character recognition is a subsequent process that transforms a bitmapped image of printed text into text code, thereby making it machine-readable.

The imaging process involves recording the changes in light intensity reflected from the document as a matrix of dots. The light/color value(s) of each dot is stored in binary digits. One bit would be required for each dot in a black/white scan; up to 32 bits would be required per dot for a color scan. The resolution, or number of dots per inch (dpi) produced by the scanner determines how closely the reproduced image corresponds to the original. The higher resolution records more information therefore, the greater file size requires higher storage and high transmission bandwidth. For example, 300 dpi achieves optimal OCR accuracy rates; 600 dpi is considered archival reproduction quality for an image, if OCR is undertaken on extremely small-font text. The image file format is typically a TIFF (Tagged-Image File Format) file. The accuracy rate is determined by the number of edits required (insertions, deletions, substitutions) expressed as a percentage of the number of characters in the image. An accuracy rate of 98% is achieved with this system.

THE OCR PROCESS

The OCR process involves the following 5 discrete phases:

- Identification of text and image blocks in the image.
- Character recognition.
- Word identification/recognition.
- Correction.
- Formatting output.

In this process, OCR generates the text, the text is saved into required format and the correction has been carried out by the user wherever required. The most common method of character recognition, called "feature extraction", identifies a character by analyzing its shape and comparing its features against a set of rules that distinguishes each character/font. Character strings are then compared with dictionaries appropriate to the language of the original. The OCR output is stored in a proprietary file format. The OCR process highlights non-recognized characters or suspicious strings and operator can make corrections if needed. After that the OCR output file is converted into one or more required formats, to give as an input to the TTS System.

TEXT TO SPEECH SYSTEM

Text To Speech (TTS) system produces the speech output for the given input text. In TTSIDL system, the main process of TTS is performed which is specified in Fig. 2.

Text normalization: The text normalization is the first step for English Text-To-Speech (TTS) synthesis (Uwe and Hartmut, 2006). It subsumes sentence segmentation, tokenization and normalization of non-standard words. Unrestricted texts include standard words and non-standard words. Standard words have a specific pronunciation that can be phonetically described either in a lexicon or by letter to sound rules. By definition non-standard words comprise numerical patterns and alphabetical strings that do not have regular entries in a lexicon and their pronunciations are to be generated.

Sentence segmentation: The main problem is the ambiguity of the period whether it marks sentence boundary or abbreviation, sometimes even simultaneously (e.g., it is 5 p.m.). For period disambiguation, an identification of abbreviation is needed. Complications arise from abbreviations that do not differ from ordinary final word of the sentence (no. also being an abbreviation of number). Ambiguation of capitalized word is another problem; Proper name and sentence initial word are to be identified properly. Another fact is proper name can occur in sentence initial position. Rule-based system for heuristic period disambiguation operates on local grammars containing abstract contexts for within-sentence periods and sentence boundaries. Machine learning approaches as the decision tree classifier, which used context features such as word length, capitalization and word occurrence probabilities on both sides of the period.

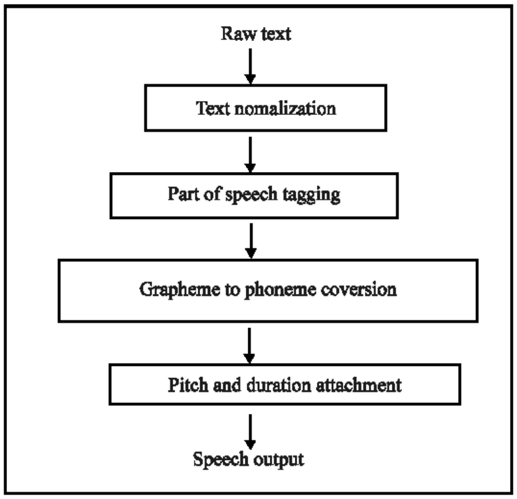


Fig. 2: Text-To-Speech conversion process of TTSIDL

Identification of proper names, abbreviations and acronyms: Since retrieval of proper names, acronyms and abbreviations is crucial for appropriate sentence segmentation and normalization of non-standard words, the identification task is carried out prior to text normalization. Due to the high productivity of these words, simple table lookup is insufficient and it has been augmented by the following procedure. Proper names, which may have one or two letters in-capitalized form, are considered as tokens. Occurrences in unambiguous environments are only counted as tokens that means, proper names which is not behind a period except for a period of prepositional titles like Mr., Dr., etc.
Token t is identified as an abbreviation, if

- It has not been classified as a proper name.
- It ends with a period.
- One of the following conditions is fulfilled:
  - *t* contains another period (*e.g.*).
  - The string of *t* preceding the period consists of just one small letter.

- *t* contains no vowel (exception *qu.*) and at least one small letter (Vs. Acronyms, numbers).
- The letter sequence of *t* indicates a violation of phonotactics.

Token *t* is identified as an acronym, if it has not been classified as a proper name or an abbreviation or a roman number and if one or more of the following conditions holds:

- *t* consists entirely of consonants.
- *t* consists entirely of capitals (except *I*).
- *t* is preceded by the article *an* and does not start with a vowel.
- *t* is preceded by the article *a* and starts with a vowel (except *u*).
- The letter sequence of *t* indicates a violation of phonotactics.

Violation of phonotactics: The phonotactics exploited here is related to the sonority-based syllable definition according to which a syllable is characterized by a sonority peak facultatively preceded by a rise and followed by a decline of sonority (for example, presence of head and coda, respectively). A letter sequence of a token indicates a violation of phonotactics if:

- The first letter is associated with a phoneme of higher sonority than that of a fricative.
- The sonority of that phoneme is higher than the phoneme associated with the following letter.
- None of the two letters in focus can be associated with a syllable.

Normalization of non-standard words: Selection of non standard word normalization (Sproat *et al.*, 2001) procedures is discussed here. Initially, the number transformations are carried out: Roman numbers are converted to Arabic numbers by calculation and arabic numbers are converted to letters by finite state transducers for cardinal and ordinal numbers. The identification of roman numbers and the distinction of cardinals and ordinals is guided by local grammars. Cardinal numbers are disambiguated whether to be pronounced as one number, as a date, or digit by digit through pattern matching and examination of the text environment regarding e.g. date-related or phone number cues. Dates are further completed by prepositions and articles accordingly. For example, 12 Feb. becomes on the February twelve, but on being omitted if a preposition is already given. After that the unknown abbreviations are spelled. The unknown acronyms are spelled which is identified by a preceeding indefinite article or by violation of phonotactics. Otherwise, they are pronounced as standard words.

Part-of-speech tagging: The POS (Brants, 2000) tagging is done by a statistical approach and can be seen as a generalization of the classical Markov tagger. According to Bayes formula, the P(w|t) emission probabilities of word w for the given tag t are replaced by a linear interpolation of tag emission probabilities given a list of representations of w, that are connected to automatically derived word suffixes. Since, in English language, suffixes also store word class information and are observed in the training data with a high probability, the Out of Vocabulary (OOV) problem can be reduced. As the approach is language independent, no linguistic knowledge is needed.

Basic form of a markov POS tagger: The aim is to estimate the probable tag sequence $\hat{T}$ given word sequenceW:

$$\hat{T} = \arg\max_{T}\left[P(T \mid W)\right] \tag{1}$$

To estimate P(T|W), a reformulation is needed by applying Bayes Formula, which leads to:

$$\hat{T} = \arg\max_{T}\left[P(T)P(W \mid T)\right] \tag{2}$$

Given that the denominator P(W) is constant. Further, the following assumptions are to be made to get reliable counts for the probability estimations:

- Probability of word w_i depends only on its tag t_i.
- Probability of tag t_i depends only on a limited tag history.

The resulting formula is thus:

$$\hat{T} = \arg\max_{t_1 \ldots t_n}\left[\prod_{i=1}^{n} P(t_i \mid t\_history_i)P(w_i \mid t_i)\right] \tag{3}$$

Generalizations of the basic model: First P(ti|t-history$_i$) is replaced by a linearly interpolated trigram model

$$\sum_{j} u_j P(t_i \mid t\_history_{ij}) \tag{4}$$

j ranging from unigram to trigram tag history. Further w_i is replaced by a list of word representations leading to a reformulation of P(w_i|t_i):

$$\frac{P(w_i)}{P(t_i)}\sum_k v_k P(t_i \mid w\_representation_{ik}) \qquad (5)$$

Again, by applying Bayes Formula and linear interpolation, the TTSIDL model is thus given by:

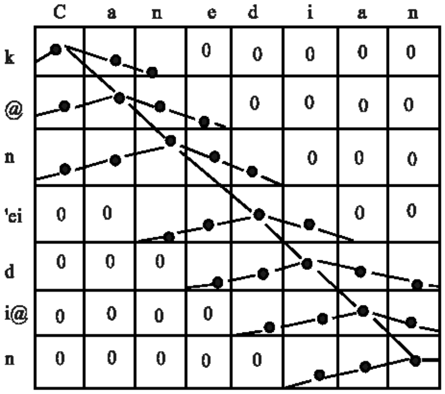
$$\hat{T} = \arg\max_{t_1 \dots t_n} \left[\prod_{i=1}^{n} \begin{array}{c} \frac{1}{P(t_i)} \sum_j u_j P(t_i \mid t\_history_{ij}) \\ \sum_k v_k P(t_i \mid w\_representation_{ik}) \end{array} \right] \qquad (6)$$

The interpolation weights u_j and v_k are calculated via the EM algorithm (Dempster, 1977). In order to reduce calculation effort in application, just for unknown words, the probabilities are calculated for all POS tags. For known words the POS tags co-occurring with the training corpus are taken into consideration.

Grapheme-to-phoneme conversion: The grapheme to phoneme approach (Yvon, 1994) is data-driven, the decision tree has been used as a classifier. The conversion is a one-to-one mapping from the set of graphemes to the set of phonemes. To cope with, any n-to-n relation the phoneme set also comprises the empty phoneme as well as phoneme clusters. A canonical dictionary is used for training and lookup. The first step for creating the grapheme-to-phoneme converter (Reichels and Schiel, 2005) is to align the phoneme string and the orthographic string of each pronunciation dictionary entry. An initial co-occurrence matrix between letters and phonemes has been estimated. It can be done by diagonally aligning the letters and phonemes of each entry. For each phoneme, a triangular window with an area of 1 and a width of 5 letters is centered at the diagonal in order to spread the probability of co-occurence to adjacent letters (Fig. 3). The values of the initial co-occurrence matrix are converted into probabilities and used in a Dynamic Programming (DP) algorithm to find the most likely alignment for each pronunciation dictionary entry.

The DP algorithm is designed to align either the empty phoneme, or one phoneme, or a phoneme cluster to each letter. Generation of sequence of phonemes for a given standard word is referred to as letter to sound rules. The complexity of these rules and their derivation depends on the nature of the language. For language such as English, a pronunciation dictionary of about 1,25,000 words is used along with a set of letter to sound rules to handle unavailable words in the dictionary used.

Prosodic analysis: The string of tagged phones enters a prosodic analysis (Antonio and Pablo, 2006) module that determines pitch, duration (and amplitude) targets for each phoneme. Prosody may even carry meaning, even in non-tonal languages (e.g., "I like the Eggs Benedict" vs. "I like the eggs, Benedict"). Therefore, prosody affects naturalness and intelligibility ratings of a TTS system. Prosody is difficult to annotate automatically. Mainly for use in manual annotation efforts, so-called Tone and Break Indices (ToBI) have emerged as the standard. Speech signal correlates the prosody which represents the duration of pauses and the pitch, as well as the phoneme durations and amplitudes. As in the case of pronunciation, prosody can be dependent on the speaker gender, on the specific speech act or application task and even on the individual speaker. This may be one reason why researchers now seem to move to use data-driven prosody approaches that employ large speech database of a single speaker over rule based system.



	C	a	n	e	d	i	a	n
k				0	0	0	0	0
@					0	0	0	0
n						0	0	0
'ei	0	0					0	0
d	0	0	0					
i@	0	0	0	0				
n	0	0	0	0	0			

Fig. 3: Initial estimation of co-occurrence values for the letters and phonemes of the word Canedian/k @ n 'ei d i@ n/. The triangular windows are used to spread the co-occurrence probability to adjacent letters

CONCLUSION

In this study, the aspects of Text To Speech Interface for a Digital Library (TTSIDL) and the modules of Text-to-speech system involved in the TTSIDL are discussed. The standards are used to validate the output of OCR and the input specification of text to speech system. The text to speech interface for a digital library provides a unique application environment to keep the textual information in electronic form and it also provides a natural sounding of speech to the corresponding text. For successful integration of TTS into the Digital library portal, there should be synergy between the user interface and the capabilities of a text to speech system. Text preprocessing module is partly data-driven. POS tagging and grapheme to phoneme conversion are partly rule-based. TTSIDL can be further improved to the other languages. In order to improve the module adaptabilities, to other languages, the amount of linguistic knowledge should be added based on the requirement. Moreover, seamless integration of speech recognition, machine translation and speech synthesis systems will facilitate the exchange of information between 2 people speaking 2 different languages.

REFERENCES

Aniruddha Sen and K. Samudravijaya, 2002. Indian accent text-to-speech system for web browsing. Proc. S_adhan_a, 27: 113-126.

Antonio Bonafonte and Pablo D. Aguero, 2006. The UPC Text to Speech Synthesis System for Spoken Translation. Proc. TC-Star Speech to speech Translation Workshop, pp: 119-204.

Brants, T., 2000. TnT-as tatistical part-of-speech tagger. In Proc. ANLP, Seattle, WA, pp: 224- 231.

Dempster, A.P. Larid, 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc., 39: 1-21.

Dan-ning Jiang, Qin Shi and Fan-ping Meng, 2006. Overview of The IBM Mandarin Text-To-Speech System. Proc. TC-Star speech to speech Translation Workshop, pp: 181-185.

Horst-Udo Hain, Jens Racky and Thomas Volk, 2006. The Papageno TTS System. Proc. TCS. speech to speech Translation Workshop, pp: 193-198.

Reichel, U.D. and F. Schiel, 2005. Morphology and Phoneme History to improve Grapheme-to-Phoneme Conversion. In Proceedings of Eurospeech Lisbon Portugal, pp: 1937-1940.

Susan Haigh, Optical Character Recognition (OCR) as a Digitization Technology. Network Notes # 37 ISSN 1201-4338, Information Technology Services National Library of Canada.

Sproat, R., A.W. Black and S. Chen, 2001. Normalization of non-standard words. Computer Speech and Language 15: 287-333.

Reichel, U.D. and H.R. Ptzinger, 2006. Text Preprocessing For Speech Synthesis. Proceedings of TC-Star speech to speech Translation workshop, pp: 207-212.

Yvon, F., 1994. Self-learning techniques for grapheme-to-phoneme conversion. Onomastica Research Colloquim, London, pp: 203-219.