NEURAL NETWORK SPEECH SYNTHESIS

Ira Gerson, Orhan Karaali, Gerald Corrigan, and Noel Massey

Speech Processing Research Laboratory
Motorola

ABSTRACT - Text-to-speech conversion has traditionally been performed either by concatenating short samples of speech or by using rule-based systems to convert a phonetic representation of speech into an acoustic representation, which is then converted into speech. Concatenative systems can require large amounts of storage, while speech from synthesis-by-rule systems may not sound natural. A time-delay neural network system is described which produces natural-sounding speech while requiring less storage than concatenative systems.

INTRODUCTION

Traditionally, text-to-speech conversion systems have relied on either carefully coded rules determining the parameters for a synthesizer — usually a formant synthesizer — or concatenation of short segments of speech such as diphones and demisyllables (Klatt, 1987; Allen et al, 1987). With the concatenative approach saving the recordings of the short speech segments can require a significant amount of storage, but the speech produced by these systems can be more natural sounding than that produced by rule-based systems. A third alternative to these traditional text-to-speech systems is to train a machine learning system, such as a neural network, to generate the mapping between a sequence of phonemes and an acoustic description from which a speech waveform can be generated (Karaali et al, 1996). It has been determined that this third approach can combine the strengths of the two traditional methods by producing high quality synthetic speech while at the same time requiring much less storage than concatenation systems.

SYSTEM DESCRIPTION

Overview

The blocks that make up the text-to-speech system are shown in Figure 1. The text-to-linguistic component, which will not be described here, converts the input text into a phonetic representation of speech, including information about prosody. Originally, this information consisted only of syntax and stress information, but a more recent version of the system uses the ToBI system (Silverman, et. al., 1992; Beckman and Hirschberg, 1994) to provide an explicit description of intonation as well. The next two components use neural networks to generate segment durations and the sequence of acoustic descriptions used by the synthesis section of a vocoder. (The corresponding analysis section of the vocoder is used to train the networks.)The duration generation, phonetics-to-acoustics, and vocoder components are described below.

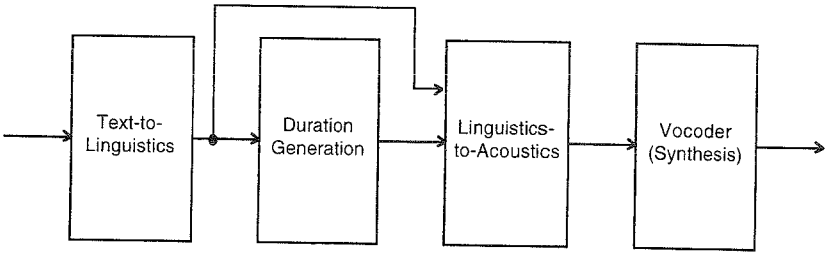

Figure 1. Block diagram of text-to-speech system during run-time

Duration network

The first step performed in converting the linguistic description provided by the text-to-linguistics component into actual speech is to establish the timing of events in the speech signal. This is performed by assigning a duration to the time segment associated with each phone in the utterance. (In this system, the timing of suprasegmental events relative to the phonetic segments is predetermined.) In order to generate each segment duration, the neural network is presented with input including: a series of binary vectors representing the phone expressed during the segment and the surrounding phones (each vector is coded with a single bit set to represent the phone); information indicating whether the phone begins or ends a syntactic constituent such as a syllable, word, phrase, clause, or sentence; binary values indicating whether the syllable containing the phone is stressed; and a bit vector based on the part of speech of the word containing the phone.

Phonetic network

Once the timing of the speech signal has been established, the phonetic network is used to convert the linguistic information and timing into acoustic descriptions of each ten-millisecond frame of speech. (The nature of these acoustic descriptions is discussed below, in the description of the vocoder.) For each speech frame, the input to the neural network includes a description of the phonetic segment which includes the start of the frame, as well as the phonetic segments including other times, sampled irregularly over a 415 millisecond window, with the sampling occurring at five-millisecond intervals in the middle of the window, and at larger intervals near the window boundaries. This use of a sampled window, rather than a sequence of phones, with the position of the frame in the phone as an additional input, contrasts with other attempts to synthesize speech with neural networks (Cawley and Noakes, 1993; Tuerk et al, 1991; Tuerk and Robinson, 1993; Weijters and Thole, 1993). Other network inputs describing the frame include: a description of the features of each phoneme (such as LABIAL or VOICED); the distance to major syntactic and prosodic boundaries; the stress of the current phone; and the word category (such as NOUN or ADJECTIVE). The phonetic network takes these input values and predicts an appropriate acoustic description for each frame.

Vocoder

The final step in generating sampled speech data from text is to convert the acoustic frame descriptions into the sampled data. This is accomplished with the synthesis portion of an analysis/synthesis vocoder. (The analysis portion is used to generate the training data for the phonetic neural network, as shown in Figure 2.) For ease of neural network training, it was decided that a parametric vocoder with a fixed frame format would be used. This vocoder used a training set of 13 parameters: ten coefficients describing an autoregressive filter for spectral shaping, the energy of the

frame, the pitch of the frame, and the boundary between two separate excitation bands in the frame. The vocoder is a mixed-source vocoder(Makhoul et al, 1978), with a low-frequency voiced band and a high-frequency unvoiced band; the boundary between the two bands is allowed to move. The output of the vocoder can be sent directly to a digital to analog converter which can be used to drive a speaker.

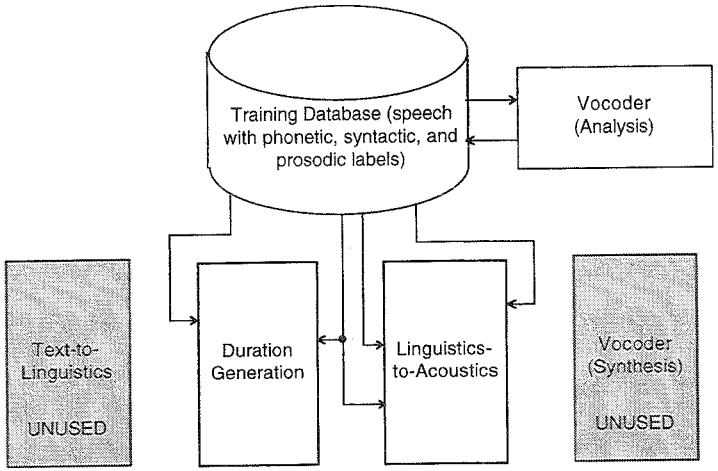

Figure 2. Block diagram of training portion of text-to-speech system

Training data

The neural networks which generate segment durations and acoustic descriptions of speech need to be trained. Figure 2 shows the system configuration used during training. The training database includes a set of recordings from a single speaker, consisting of approximately twenty minutes of speech. This speech was labeled in a variety of ways. The original phonetic labeling was performed in the manner described by Seneff and Zue (1988). The speech was also divided into syllables, words, phrases, clauses and sentences. Each syllable was tagged as having no stress, secondary stress, or primary stress. Each word was assigned a tag based on the part of speech of the word. Finally, the files were labeled according to the ToBI annotation conventions, as described by Beckman and Hirschberg (1994). The speech was processed using the analysis portion of the vocoder and combined with the label data to provide a set of training vectors containing input and desired output for the phonetic network. The network was then trained using back propagation. The label data was also processed to produce training data for the duration network, which was also trained using back propagation.

RESULTS

In an independent comparison of this speech synthesis system with other text-to-speech system (Nusbaum, et al, 1995), this system was found to have a more acceptable voice to listeners than the existing systems. The neural network system was found to be less intelligible than the other systems. In part, this may be because the network was only trained on sentence-length texts, and the intelligibility test used monosyllables spoken in isolation. The database is being expanded to include examples of monosyllables, as well as a greater variety of sentence styles.

Figure 3 shows the results for the acceptability test. The listeners were asked to rate the quality of speech in a variety of sentences they heard, with one representing unacceptable speech, and 7 representing speech they would want to hear again. The systems compared were three traditional text-to-speech systems, including one concatenative system and two synthesis-by-rule systems. The two neural network systems included one in which durations were matched to natural speech (NN 1) and one with the durations generated by a neural network (NN 2). The input text to the text-to-speech systems was adjusted to provide correct pronunciations, as the experiment was intended to compare the speech synthesis technologies, not the overall performance including text analysis. The neural network systems performed significantly better than any of the traditional systems in this test.

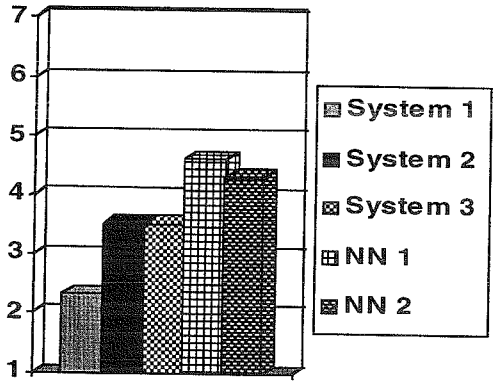

Figure 3. Mean acceptability score for various systems

Figure 4 shows the results from the segmental intelligibility test. In this test, listeners were asked to identify monosyllabic words generated by the systems. The test measures how many of these words were identified correctly. The same systems were compared as in the acceptability test, and natural speech was also included. As the result show, two of the text-to-speech systems had significantly greater segmental intelligibility than the neural network systems, but natural speech was more understandable than any of the systems.

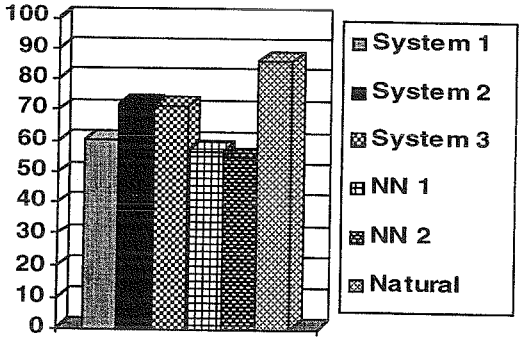

Figure 4. Percent of words recognized correctly by various systems

The system has been implemented to run in real time on both a Power Macintosh 8500/120 and a Windows 95 system using a 200 Mhz. Pentium Pro microprocessor.

CONCLUSION

Text-to-speech synthesis based on neural networks is demonstrated to provide the benefit of generating more natural sounding speech than traditional methods. Real time implementation of the system demonstrates the feasibility of the technology. Further benefits expected for the technology is the relative ease of providing new languages and voices for the text-to-speech system.

REFERENCES

Allen, J.; Hunnicutt, M. S.; and Klatt, D. (1987). "From Text to Speech: The MITalk System" (with R. C. Armstrong and D. Pisoni), Cambridge: Cambridge University Press.

Beckman, M. E. and Hirschberg, J. (1994). "The ToBI annotation conventions", In ToBI release 2.0[database online]. New Mexico Institute of Mining and Technology. Identifier /tarfiles/tobi_release_2.0.tar.Z Available from kiwi.nmt.edu.

Burniston, J. and Curtis K. M. (1994). A Hybrid Neural Network/Rule Based Architecture for Diphone Speech Synthesis. In *International Symposium on Speech, Image Processing and Neural Networks Proceedings*. 323-6.

Cawley, G. C. and Noakes P. D. (1993). "LSP speech synthesis using backpropagation networks", In *Third International Conference on Artificial Neural Networks Proceedings*. 291-4.

Karaali, O.; Corrigan, G.; and Gerson, I.; "Speech synthesis with neural networks", In *WCNN 96 Proceedings*, 45-50.

Klatt, D. H. (1987). "Review of text-to-speech conversion for English", J Acoust. Soc. Am. 88, 737-793.

Makhoul, J.; Viswanathan, R.; Schwartz, R; and Huggins, A. W. F. (1978). "A mixed-source model for speech compression and synthesis", In *ICASSP 1978 Proceedings*, 163-166.

Nusbaum, H; Francis, A.; Luks, T. (1995). "Comparative Evaluation of the Quality of Synthetic Speech Produced at Motorola", Technical Report 1, The University of Chicago.

Seneff, S. and Zue, V. (1988). "Transcription and alignment of the TIMIT database", Photocopy. (Originally distributed with the TIMIT database from NIST.)

Silverman, K.; Beckman, M.; Pitrelli, J.; Ostendorf, M.; Wightman, C.; Price, P.; Pierrehumbert, J; and Hirschberg, J. (1992). "ToBI: A standard for labeling English prosody", In *ICSLP 92 Proceedings*. vol 2, 867-870

Tuerk C.; Monaco, P.; Robinson, T. (1991). "The Development of a Connectionist Multiple Voice Text To Speech System" In *ICASSP 91 Proceedings*. vol 2, 749-52.

Tuerk, C. and Robinson, T. (1993). "Speech Synthesis Using Artificial Neural Networks Trained on Cepstral Coefficients", *In Eurospeech 93 Proceedings*. vol 3, 1713-6.

Weijters, T. and Thole, J. (1993). "Speech Synthesis with Artificial Neural Networks", In *ICNN 93 Proceedings*, vol 3, 1764-9.

MAXIMUM A POSTERIORI DECODING FOR SPEECH CODEC PARAMETERS

W.N. Farrell and W.G. Cowley

Institute for Telecommunications Research
Department of Electronic Engineering
University of South Australia

email : wade@spri.levels.unisa.edu.au

ABSTRACT – This paper looks at applying MAP decoding to low rate speech codec parameters as a means of protection at low channel SNRs. It has been shown that MAP decoding works well in protecting LSPs but the method has not been applied to other parameters. By using theoretical source data, results are obtained that compare MAP decoding with other more conventional techniques.

INTRODUCTION

When considering the use of a parametric low rate speech codec for a speech transmission system, the normal method of design involves using separate source and channel coding stages. The design goal is thus to compress the source information as much as possible and then apply a good channel codec to that information, given all the system and channel constraints. This philosophy is valid if all of the redundancy is removed from the source information. Unfortunately rarely does all redundancy be removed from the source information, especially when considering speech codec parameters.

An alternative method is to leave the residual redundancy present in the coded speech information and use the redundancy at the channel decoder. In this way all of the redundancy can be used for channel protection instead of trying to extract it by using complicated quantisation techniques.

The aim for such a channel decoder is to utilise all information about the signal, which involves both channel information and prior information about the source. This paper provides a formal analysis using Bayesian probabilities to describe a maximum a posteriori probability (MAP) decoder. It also explores how such a decoder can be used to protect low rate speech codec information. Two simple test cases are examined using Gaussian and Gauss–Markov source parameters.

MAXIMUM A–POSTERIORI DECODING

To construct a decoder, a model of a communication system is required for analysis. The model in Figure 1 is a simplified case, consisting of a speech codec providing speech parameter value X at time index n. This is channel coded and modulated to create symbol U_n. This symbol is passed through the channel, where noise is added to the symbol and then received as V_n ($V_n = U_n + Noise$). This paper assumes the use of an Additive White Gaussian Noise channel. After decoding, an estimate of the original value is obtained. This scalar case is used to simplify the analysis, however sequences are explored later.

Figure 1 – Simple Speech Communication Model

Using the MAP model above, the estimator in equation (1), as described in [1], states that the source value X_n can be estimated as the expected value of X_n given the received value of V_n.

$$\hat{X}_n = E(X_n|V_n) \tag{1}$$