REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1995	Professional Paper

4. TITLE AND SUBTITLE

TEXT-TO-SPEECH PHRASING ENHANCEMENT SYSTEM USING NEURAL NETWORKS

5. FUNDING NUMBERS

PR: CDB8
PE: 0602233N
WU: DN309119

6. AUTHOR(S)

L. F. Julig

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152–5001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Much progress has been made in computer text-to-speech systems over the past several years. In particular, the Macintosh computer systems now provide the PlainTalk™ Text-To-Speech synthesizer which is capable of using high quality voices with various attributes to convert text to synthesized speech. Though these new voices and speech synthesizer are great improvements over the previous Macintalk™ system, the synthesized speech still sounds far from natural. One attribute which could add greatly to the naturalness of the speech is improved phrasing. The PlainTalk™ Text-To-Speech synthesizer provides the means to embed speech commands within text to modify the spoken output. The purpose of this project is to build a neural network which through supervised learning will produce an algorithm for embedding pitch controls in text which will produce more natural sounding emphasis variations for the spoken output.

Published in *Proceedings of the Society of Women Engineers 1995 National Convention Conference*, pp. 60–65, July 1995.

14. SUBJECT TERMS

Neural Networks
Computer Text-to-Speech
Synthesized Speech

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b. TELEPHONE *(include Area Code)*	21c. OFFICE SYMBOL
L. F. Julig	(619) 553–7863	Code 44215

TEXT-TO-SPEECH PHRASING ENHANCEMENT SYSTEM USING NEURAL NETWORKS

Louise Frantzen Julig, Electrical Engineer

Naval Command, Control, and Ocean Surveillance Center
Research, Development, Test & Evaluation Division Code 44215

53355 Ryne Rd. Rm. 101
San Diego, CA 92152-7252
(619) 553-7863 ph. (619) 553-9229 fax
julig@nosc.mil

ABSTRACT
Much progress has been made in computer text-to-speech systems over the past several years. In particular, the Macintosh computer systems now provide the PlainTalk™ Text-to-Speech synthesizer which is capable of using high quality voices with various attributes to convert text to synthesized speech. Though these new voices and speech synthesizer are great improvements over the previous Macintalk™ system, the synthesized speech still sounds far from natural. One attribute which could add greatly to the naturalness of the speech is improved phrasing. The PlainTalk™ Text-To-Speech synthesizer provides the means to embed speech commands within text to modify the spoken output. The purpose of this project is to build a neural network which through supervised learning will produce an algorithm for embedding pitch controls in text which will produce more natural sounding emphasis variations for the spoken output.

Introduction

Efforts to synthesize speech mechanically have been underway since before the days of Alexander Graham Bell, and computer speech synthesis technology has been available since the 1960s. With the advent of linear predictive coding and formant synthesis in the 1970s, computer speech synthesis started to become more common. Improving the naturalness of synthetic speech has been an ongoing process since then. Obviously, simply stringing together sequences of phonemes and syllables does not produce natural or even intelligible speech - some method of indicating prosodic, or intonational, stress patterns must be used (Linggard). Much work has been done in the grapheme to phoneme realm, and most commercially available speech synthesizers today produce very natural sounding words, incorporating appropriate lexical stress patterns within each word. Little work has been done, however, in attempting to produce natural sounding phrases made up of these words. One characteristic of natural speech which contributes to the perception of phrasing is pitch accent. Pitch accent is the perceived stress on a word as it is spoken in an utterance as opposed to the characteristic lexical stress pattern of each word (Hirschberg). The ability to accurately predict these pitch accent patterns for unrestricted text is currently unavailable, which allows for considerable research opportunities. As Hirschberg notes, "How humans decide which words to accent and which to deaccent - what constrains accent placement and what function accent serves in conveying meaning - is an open research question in linguistic and speech science."

The goal of the Text-to-Speech Phrasing Enhancement System is to learn pitch accent patterns from sample utterances, extract generalized rules about the placement of pitch accent within utterances, and apply the rules to the output of a text-to-speech system. For a first pass implementation of this system, a more refined goal of learning pitch patterns from grammatical parts of speech using neural network technology was proposed. The justification for simplifying the input text to grammatical part of speech is given by the fact that recent work in the area of pitch accent prediction (Altenberg, Hirschberg), has indicated that minimal text analysis such as that provided by a grammatical parser may be able to provide a great deal of prosodic information. The simplification of pitch accent prediction to

just pitch prediction is justified by the fact that of the three physical attributes that the perceptual phenomenon of accent or stress can be considered to have: pitch, phoneme duration, and speech amplitude, it has been shown that, within sentences, pitch changes are the main contributors to the subjective impression of stress (Linggard). The choice of neural network technology was made because the characteristics of neural networks make them particularly suited to this type of problem and because the work on this project was completed as part of a course in neural networks. These characteristics will be discussed in a brief overview of neural networks.

Neural Networks

Artificial neural networks, commonly referred to as "neural networks," can be defined as parallel, distributed, adaptive information processing systems that develop information processing capabilities in response to exposure to an information environment (Hecht-Nielsen). They have several characteristics which make them particularly appropriate for use in many speech and language related problems. Some of these characteristics are: 1) Nonlinearity - A neural network is made up of connections of individual neurons, which are in themselves nonlinear. Thus the nonlinearity is distributed throughout the system and is an important property if the underlying physical mechanism responsible for the generation of an input signal, such as speech or language information, is inherently nonlinear (Haykin). 2) Pattern Association - A neural network is said to associate pattern A with pattern B if it produces an output pattern of activity *A* in response to an input stimulus pattern *B* (Morgan). This characteristic, also known as input-output mapping, is extremely important in the learning process. What makes this characteristic so important is the fact that the patterns which are being associated do not necessarily need to be known in advance, but can be extracted from the data by the network. 3) Generalization - The reproduction of a single output pattern of activity, despite distortions in stimulus pattern due to natural variability, noise, or missing information, is the characteristic of generalization (Morgan). This characteristic is what enables a neural network to be trained on a sample set of data and then apply the input-output mappings it has learned to entirely new data sets. This characteristic is extremely important to any real life system in which the network could never practically be exposed to all possible input stimuli.

Neural network approaches to various speech recognition and speech synthesis problems have been used since the mid to early 1980s. One of the first demonstrations of the use of a neural network for speech synthesis was the NETtalk text-to-speech system developed by Sejnowski and Rosenberg. In this system a neural network was presented with a sliding window of seven "letters" from sample text, including spaces and punctuation marks. For each window, the object was to learn what phoneme to pronounce for the middle letter. The network was trained using supervised learning from a rule-based speech synthesis system to produce the correct phonemes. The success of NETtalk was important in that it showed that complex linguistic information can be learned and represented in a neural network structure. Other work has focused on using neural networks to learn fundamental frequency contours and prosodic information for individual words (Karjalainen, Scordilis). This project's approach slightly differs from these works in that it attempts to learn "macro-pitch contours" over several words in a phrase rather than the pitch information within a single word or phoneme.

Design

The concept behind the Text-To-Speech Phrasing Enhancement System (hereby known as the enhancement system), is to encode input text into separate grammatical elements, relate those elements to a desired macro-pitch contour, and use a neural network to learn the input-output mapping between the grammatical structure and the desired pitch when the text is spoken aloud. This mapping will then be applied to new text examples, generalizing to accommodate new patterns in grammatical structure. The output will be a pitch contour which can be applied to the input text to produce a "phrased" utterance. The methodology in designing the system will be discussed next.

The architecture of the enhancement system is based closely on the NETtalk design. While the goals of the two systems are not the same, they are similar in that they both use a pattern

recognition approach to learn complex linguistic information, and the approach of the NETtalk system applies readily to the enhancement system. The neural network is presented with an odd number of inputs and the object is to predict some attribute of the center input. In the enhancement system, the inputs are five word elements, where a word element is any word or piece of punctuation. The desired output is the pitch for the third or center word element. A simplified representation of this structure is shown in Figure 1.

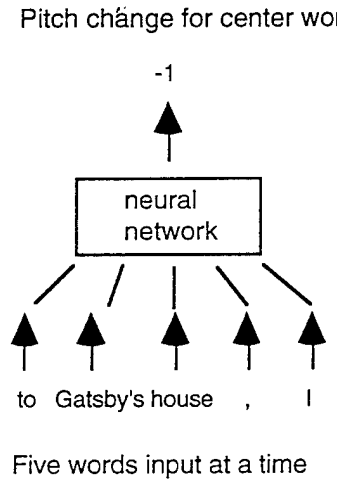

Figure 1: Basic system structure

In order to implement the system in Figure 1, first the training and testing sample text needed to be chosen. Next the inputs and desired outputs of the network needed to be encoded in a manner conducive to the learning process of the network. The selection of the training examples and the coding of the inputs and the outputs will be discussed below.

Training examples

The text samples which were used consisted of two passages of four or five sentences each from three different works of fiction. The works chosen were *The Plague*, by Albert Camus, *The Great Gatsby*, by F. Scott Fitzgerald, and a short story entitled *Hope*, which was posted on the America Online service. Fiction works were chosen because the phrasing of a fiction work read aloud is likely to be more distinctive than a work of non-fiction read aloud.

The first step in preparing the data was to encode the input text by part of speech. Each word and piece of punctuation in the text samples was tagged with a part of speech using the commercial grammar checking software Grammatik™. The parts of speech were then grouped into nine categories: noun, verb, pronoun, adjective, adverb, preposition, conjunction, comma, and other punctuation including periods, semicolons, and colons. Each word element was then coded with a one out of nine code corresponding to its part of speech. For example, a noun was coded as "100000000", a verb was coded as "010000000", and so forth. This "one out of n" coding is done to ensure that each input category can be represented as a unit vector which is orthogonal to all other input category vectors in an n-dimensional space. This aids the learning process by providing data which is already linearly separable. Thus, the input to the neural network for any given trial is a sequence of five word elements coded in this way, or 9 x 5 = 45 digits, of which one in every nine is a 1 and the rest are 0's.

Once the inputs were encoded, the next step was to prepare the desired output training data by encoding the desired pitch contour for a text segment in a way that could be presented to the network. Since the Plaintalk™ speech synthesizer provides the capability to embed pitch commands within text to modify the spoken output, it was decided to create the desired pitch contours by annotating the sample text with these pitch commands. The embedded commands were tuned manually using subjective judgment until the experimenter deemed that the synthesized speech sounded sufficiently natural with regard to the pitch changes. A sample of some annotated text is given in Figure 2. The pitch commands are given in terms of units which have a fixed relation to frequency in Hertz (Apple). The

[[pbas -1]] I [[pbas +4]] believe [[pbas -3]] that on the [[pbas +2]] first [[pbas -1]] night [[pbas -1]] I [[pbas +2]] went [[pbas -2]] to [[pbas +1]] Gatsby's [[pbas -1]] house I was one of the [[pbas +4]] few [[pbas -4]] guests who had [[pbas +3]] actually [[pbas -1]] been [[pbas -1]] invited .

Figure 2: Example of annotated text

embedded command "[[pbas -1]]" tells the speech synthesizer to lower the pitch by one unit below the baseline until another command is reached. When the next command is reached, "[[pbas +4]]" in Figure 2, this tells the speech synthesizer to raise the pitch 4 units from the previous level. Thus each command changes the pitch relative to its immediate previous level. Since a pitch command reflects only a change from the previous pitch, any given sequence of five word elements and associated pitch change annotations taken out of context gives no indication of a reference to the original baseline pitch. For this reason, pre-processing software was developed which converts the sequence of pitch changes to an absolute pitch value relative to the baseline pitch. Any group of five word elements and associated pitch change annotations taken out of context could thus still be referenced to the baseline pitch. These absolute pitch values were then assigned a one out of eleven code for each word element in the training text. This one out of eleven code is similar to the one out of nine coding used for the input data. Ten of the eleven digits are 0 and one is a 1. The location of the 1 depends on the number of pitch units above or below the baseline pitch for that particular word, within a range of +/- 5 from the baseline. For example, a pattern of "00000100000" indicates the baseline, and a pattern of "00000001000" indicates 2 pitch units above the baseline.

Neural network architecture

The network which was chosen to implement the enhancement system is the multi-layer perceptron (MLP), or backpropagation architecture. This network was chosen for two reasons. First, the MLP is well suited to problems of pattern recognition, of which this problem is one, and second, it is the same network which was used for the NETtalk system, to which the enhancement system has many parallels. Figure 3 shows a simplified picture of the MLP neural network which was used for this project. For a description of the MLP backpropagation network, see any number of neural network references (Demuth, Haykin, Hecht-Nielsen). The network was implemented using the Neural Network Toolbox of the MATLAB® software package by The MathWorks. A three layer network with one hidden layer of 25 processing elements was

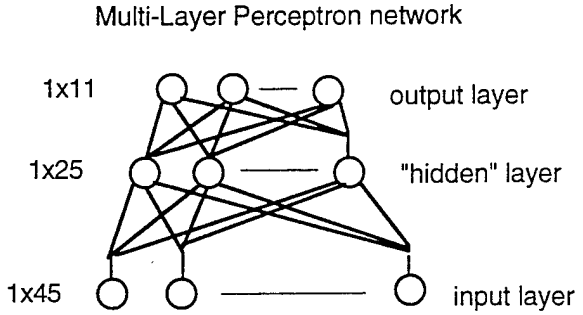

Figure 3: Neural network structure

used. The input and output layer sizes are dictated by the data, and the size of the hidden layer was chosen to be approximately midway between that of the input and output layers. The hidden layer and output layer used a log-sigmoid transfer function.

Training and testing

Of the six text passages used, one passage from each work was used for training and one for testing, for a total of three training passages and three testing passages. To train the network, a window of five word elements, each encoded with its one out of nine part of speech code, is entered into the network. The training method used by the network was fast backpropagation using momentum and an adaptive learning rate. The network then produced an output vector of eleven elements, the goal of which is a one out of eleven code representing the target pitch for the center word of the input. During this type of training phase the network then reads the target vector, calculates the error between the network output and the target output and "back-propagates" this error. Each time this process is repeated, the weights of each processing element are changed to produce an output that is closer to the target output on the next training episode. One input-target pair results from each word element in the training passages, for a total of 264 training pairs. Before the network was trained, the pairs were randomly permuted so that they were not presented to the network in sequential order, and the starting weights of the processing elements were initialized to small random values. During the training phase, the network can either be told to train for a set number of epochs (an epoch is one complete run through all training examples) or until the sum squared error between the network output and

the target output falls below a certain threshold (error goal). The network can be told to stop training after a set number of epochs if the error goal has not been reached. The enhancement system network was trained with various numbers of epochs towards a sum squared error goal of 1. After several training attempts, it was found that after approximately 1000 epochs the mean squared error leveled off to a value of 38.84. At this point the weights of the processing elements were frozen and the network was ready for testing. During the testing process, encoded input text is presented to the network and an output vector is produced. The output can be compared to the desired output, if it is known, to measure the effectiveness of the network but no further training is done.

To test the network after the weights have been frozen, a meaningful test metric needed to be determined. A standard measurement is the mean squared error between network output and desired output, but for this particular problem, the mean squared error did not seem to be meaningful. Thus, another error measurement was needed. Since the elements of the network output vector are floating point numbers, the first step was to convert this output vector to a true one out of eleven code. This was done by simply choosing the maximum value in the output vector and setting it to 1, and setting the other ten values in the output vector to 0. Next an error measurement was devised which counted up the total number of times the network produced the correct output, meaning the 1 was in the correct position among the ten 0's. Counts were also taken of how many times the position of the 1 in the network output was off by a distance of one, two or three, four our five, and greater than five units from the position of the 1 in the desired output. The reasoning was that if the pitch goal is off by only one or possibly two units, the resulting speech might still be acceptable. With this error measurement, the network produced the correct value for the training data 83.4% of the time, and was only off by one position 4.5% of the time. For the new non-training data, the network produced the correct value only 13.1% of the time, but was only off by one position 29.17% of the time, and was off by two or three positions 25.21% of the time. The network produced outputs which were off by more than three positions 35.52% of the time.

Results

For any system designed to improve the naturalness of synthetic speech, the only real method for judging the quality of the system is to listen to the results. For this project, the results were mixed. The network produced very acceptable output for the text passages upon which it was trained, which is expected. As noted, the network produced the correct pitch value for 83.4% of the words in the training corpus, and the resulting synthesized speech has pitch phrasing which, while not entirely natural, is an improvement over the monotone reading one receives when no pitch change information is given at all. For the non-training text passages, the results were not as consistent. The resulting speech has some pitch changes which sound odd, but others produce a natural-sounding twist on the pronunciation. The pitch contours were different from the way the experimenter annotated the text, but a valid alternative nonetheless.

Two additional factors affect the way the results of this project are viewed. They are a modulation control and ending prosody built in to the PlainTalk™ voices. The modulation control sets a modulation range about the baseline pitch. The result is a speech channel that does not speak in a monotone, but instead continually varies the pitch about the baseline. Ending prosody applies the speech tone and cadence that normally occur at the end of a statement when the text-to-speech synthesizer encounters a period or other similar punctuation. The result is speech that naturally falls at the end of sentences instead of simply pausing. These two parameters combined do a lot to improve the naturalness of the PlainTalk™ speech synthesizer without the phrasing enhancement system. This is not to say that the results of the enhancement system project are not of any use - with some improvements the enhancement system may well be able to provide even more phrasing help to the already built-in functions of the speech synthesizer. Even as the system exists now it can be useful, because it is much easier to fine-tune the already embedded pitch controls than to produce them from scratch.