A COMPARISON OF ENERGY-BASED ENDPOINT DETECTORS FOR SPEECH SIGNAL PROCESSING

Kevin Bush, Aravind Ganapathiraju, Paul Kornman, Jim Trimble III, Leigh Webster

Speech Processing Group
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
{bush,ganapath,trimble,webster}@isip.msstate.edu

ABSTRACT

Accurate endpoint detection is ia necessary capability for efficient construction of speech databases based on field recordings. The labor required to prepare a database often exceeds the amount of speech data by two to three orders of magnitude. The associated cost prevents many historical databases currently stored on analog or digital tape from being made accessible, in a useful form, to speech researchers. More recent algorithms based on Hidden Markov Models (HMMs) and Neural Networks (NNs) are not always an efficient method of segmenting such data, since they are complex and highly channel dependent. In this paper we describe the implementation of a family of endpoint detection algorithms which uses signal features based on energy and zero-crossing rate. We also present a detailed comparison of these widely used algorithms using an objective evaluation paradigm we have developed. A reference speech database has been created as support for this evaluation methodology. Our implementation makes extensive use of object-oriented concepts and data-driven programming techniques. A uniform user-interface for all algorithms is provided that is based on a novel virtual class methodology.

1. INTRODUCTION

A major source of errors in speech recognition systems is the incorrect selection of the beginning and ending of speech utterances. A fundamental aspect for these algorithms is that speech segments must be reliably separated from non-speech segments. Because the endeavors to adjust these incorrect beginning and ending points do not always succeed, robust word begin and end point detection under unfavorable conditions still remains an unsolved issue because speech endpoint detection is trivial when used under ideal conditions; a simple energy calculation can be used. In recent studies, it was shown that in a real-world evaluation of a speech system which utilizes an isolated-word recognizer more than 50% of the error rate was credited to the endpoint detector. [14] According to Savoji, [35] the essential characteristics of an ideal endpoint detector are: reliability, robustness, accuracy, adaptation, simplicity, real-time processing, and no prior knowledge of the noise. Of all of these characteristics, robustness in unfavorable conditions has been the most difficult to accomplish.

Another problem in speech recognition is the high computational load on the system. With the detection of inaccurate begin and end points, wasteful computations will be made. In order to operate in real-time, speech detection algorithms must be efficient both in terms of the speed of the computation and the memory consumption. Also, the input signal must be buffered since the exact start and end of live input is unknown. This system of buffering allows the real-time endpointer to run in parallel with the live input.

Several methods have been used for accurate endpoint detection. Among these methods are zero-crossing rate[3], energy distribution [2], spectral information, periodicity measures [4], and hidden Markov models (HMM) [5].

2. SEGMENTATION

In endpoint detection, one of the most difficult aspects is

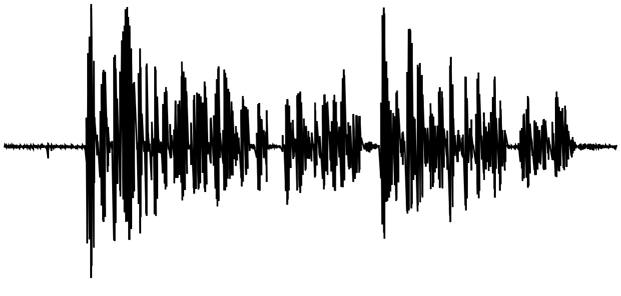

Figure 1. Speech Waveform.

segmentation of utterances. Manual segmentation is an option, but there are several drawbacks. The first drawback is that the process is laborious and tedious. It requires extensive listening and spectrogram interpretations. Also, due to the subjective nature of a manual segmentation, inconsistencies from trial to trial, even if the same utterance is being segmented, also hinder the process. Throughout the past decade, several methods for automatic segmentation have been proposed. Finally, van Hamert[37] described an automatic segmentation method that combines explicit information about the speech with the frame-to-frame spectral change. The frame-to-frame spectral change is in part characterized by a spectral correlation function.

In many commercial recognizers, word units are used and speakers are instructed to pause briefly after each word which leads to isolated word recognition. A silence 150 and 250 ms between words is sufficient enough so that the utterance is not confused with long plosives and allows the recognizer to compare words rather than sentences. In recent years, recognizers have been constructed to reach a capability of recognizing connected-word speech. Connected-word speech implies that there are no pauses required, but each word must be pronounced clearly. Because of the unnatural speaking constraints of isolated words, many applications of connected-word recognition can be considered efficient as in postal codes, telephone numbers, and spelled-out words. Connected-word recognition is simpler than continuous-speech recognition because clear pronunciations and pronounced stressing of each utterance help minimize the effects of coarticulation across word boundaries.

Typical endpoint detectors depend on amplitude functions to separate nonspeech from speech. One method for finding endpoints locates energy pulses corresponding to words or syllables and comparing the energy in decibels against four thresholds k1, k2, k3, and k4. [4,32] When the energy exceeds the lowest threshold, k1,a pulse is considered to be detected starting at A1, unless the duration A2-A1 exceeds 75 ms, A2 is then seen as the start time and the signal before A2 is considered breath noise). The end time is similarly determined by means of k2 and k3 (5dB).Successive energy pulses may be considered part of one unit if the gap time between pulses is less than 150 ms, the longest duration allowed of a typical stop closure.

For those applications where the speech bandwidth exceeds 3 kHz, sufficient spectral information at high frequencies is present to refine energy-determined boundaries with simple spectral measures [1]. The zero-crossing rate provides a simple spectral measure of the frequency in the middle of the signal bandwidth. For other speech obstruents, the zero-crossing rate, if the voicebar dominates, is either low or high. The zero-crossing rate is high when weak fricatives are present. Weak fricatives also cause the most detection difficulties.

2.1. Segmentation of Connected-Words

Defining boundaries within speech segments is extremely difficult. Most recognizers erroneously make the assumption that speech is linear and invariant. The use of isolated words avoids accounting for the effects of coarticulation that obviate these assumptions. Coarticulation cannot be ignored when continuous speech is used. Segmenting connected-word speech into words is feasible only when each word is clearly spoken and the vocabulary is very limited.

One approach for continuous word recognition that can be applied to words of arbitrary length involves a speaker-independent, statistical approach. The segmentation process is followed by isolated word recognition on the separated words.

2.2. Segmentation of Continuous Speech

The convex hull method [36] is an easier segmentation approach that is not hindered by connected-word speech. Since there are few acoustic cues to distinguish word and syllable boundaries without feedback from higher levels, the boundaries obtained with this vocabulary-independent procedure are for syllables, not words. It should be noted that the significance of an energy dip for segmentation depends on the energy in the surrounding segments. After speech segments between pauses have been isolated, a convex hull is determined from the speech loudness function, a perceptually weighted energy vs. time plot that is lowpassed to eliminate pitch period effects. A convex hull exhibits minimal magnitude, monotonically nondecreasing until the loudness peak and monotonically nonincreasing thereafter. The depth of each loudness valley under the hull indicated the possibility that a boundary is present. If the maximum depth exceed 2 dB, a boundary is declared and the process is repeated with new hulls on each side of the boundary. Because a loudness change is more noticeable at syllable onset than offset, syllable-initial boundaries are more accurately located than syllable-final ones.

2.3. Segmentation of Continuous Speech into Phones

One of the most difficult problems in continuous speech recognition is reliably dividing continuous speech into phones. No specific approach exists. Typical speech parameters and features are useful for labeling the segments. The sequence of segmenting and labeling is difficult subject. Because phone segments contain several frames and labeling a phone often requires examination of the spectral behavior of the course of the phone's frames, it is more efficient to segment first. Still, some systems label each frame independently before segmentation despite the higher computational load involved. Usually, Continuous Speech Recognition attempts a coarse segmentation first and then refines the boundary placements during the labeling phase. The initial segmentation goes beyond the syllable division described above to smaller units such as fricatives, stops and vowels. These units can then be more readily located with robust and simple tests involving bandpass energies, zero-crossing rates, and durations. Dynamic programming is used to overcome errors by aligning phonetic labels with estimated boundaries.

Even though continuous speech recognition systems use formants for labeling, a coarse segmentation can be obtained using more reliable features. In this case, the spectrum is typically divided into four regions. Some of these regions roughly correspond to formant ranges. However, errors such as missed formants are avoided by using broad energy measures in each frequency range. For example, there will typically be a voice range of about 80 to 250 Hz, a low range of about 300-1000 Hz, a middle range of 700-2500 Hz, and a high range of above 2500 Hz.

Some continuous speech recognition systems also use a a four category initial segmentation for vowels, silences, fricatives and dips. Silence can then be determined by segments longer than a minimum duration whose energy above 300 Hz falls below some threshold that is normally set about 3 dB above the background noise level. A vowel would include voiced segments longer than some duration with more energy at high frequencies than low frequencies. After a silence, a frame could be checked for brief energy burst in the high range. If this burst were greater than a certain time frame, this would fall into the fricative category. If it were less than that range, it could be a stop burst or a stop aspiration. A dip would then be an energy drop of more than about 60% relative to adjacent energy peaks. Nasals are easily identified by mid-range dips. Dip detectors usually smooth the energy parameters over a few frames before segmentation is attempted.

3. HIDDEN MARKOV MODELS

Hidden Markov models are widely used as speech input unit models in speech recognition and utilize a stochastic model of speech production while offering performance comparable to time warping in several applications at a fraction of the computational cost. Training performed on a large speech database determines HMM parameters. The method of HMMs incorporates a system that exists in a finite number of varying states to model speech generation. Each varying state can produce a finite number of outputs. In word generation, the system moves from one state to another while each state creates an output until the entire word is produced. Figure 2 illustrates this process. In the figure, states are represented by circles, and arrow represent transitions between states. The transitions between the states and the outputs of each state are random which allows the model to handle subtle variations in timing and pronunciation.

Each word is represented by a model of this kind. The only item a speech recognizer has to work with are the outputs. The primary job of the recognizer is to decide which model produced the desired output. The model itself is not visible to the recognizer. It is inferred from the available data hence the word *hidden* in the name hidden Markov model.

In transitions between states, the assumption is made that they happen at discrete times and that each transition from state q_i to state q_j has a probability dependent only on state q_i. In Figure 2, respective probabilities are written by the arrow that represents the transition. An NxN matrix **A** containing N states where a_i=P{transition from q_i to q_j} is shown below:.

$$A = \begin{bmatrix} 0.3 & 0.5 & 0.1 & 0 & 0.1 \\ 0.2 & 0.4 & 0.4 & 0 & 0 \\ 0 & 0.1 & 0.3 & 0.5 & 0.1 \\ 0 & 0.1 & 0.1 & 0.5 & 0.3 \\ 0.2 & 0 & 0 & 0.2 & 0.6 \end{bmatrix}$$

It should be noted that the probabilities of the outgoing transitions from any state sum to 1; therefore, each row of **A** must sum to 1. By making the transitions nondeterministic, the model can handle deletions or repetitions of states, which is a highly desirable capability. The system also has more than one starting

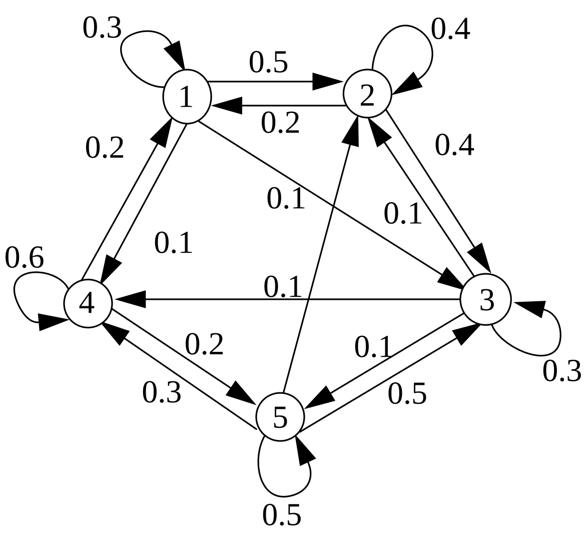

Figure 2. HMM State Diagram.

state. p(1) represents the initial state probabilities, where $p_i(1)$=P{initial state q_i}.

For a finite set of M possible outputs {z_i}, each state q_i has an M vector b_i (b_{ij}=P{output z_j | state=q_i}. The outputs of all the states can then be represented by the NxM matrix **B** whose ith row vector is biT. Also, each row of **B** has to sum to 1 since the output probabilities for each state must sum to 1.

This model also assumes a finite number of discrete outputs; therefore, in a continuous signal, e.g. a speech signal, a method has to be discovered that selects reasonable prototype outputs for {z_i}. Rabiner *et al.* (1983) solved this issue through the use of vector quantization. Vector quantization automatically results in a desirable clustering where each cluster is associated with an output z_i. Rabiner *et al.* also found that recognition performance with a left-to-right model, as shown in Figure 3, was far superior to that of less-constrained forms. In a left-to-right model, an earlier state cannot be returned to because there is a single starting and a single ending state.

With the probabilities, a discrete-time, discrete-state Markov process exists. At any time t, the probability of entering q_j from q_i is equal to the probablity of having been in q_i at time (t-1), times the probablity of transition. To find the over-all probability of being in q_j, we sum these products over all possible previous states:

$$p_i(t) = \sum_{i=1}^{N} a_{ij} p_i(t-1)$$

pi(t) represents the ith element of the vector p(t). The probabilities of all the states at time t can be shown by

$$p(t) = Ap(t-1) = A^{t-1}p(1)$$

Similarly,

$$\begin{aligned} p\{z_k\ at\ time\ t\} &= \sum_{i=1}^{N} b_{ik} p_i(t) \\ &= b^T{}_k\ p(t) \\ &= b^T{}_k\ A^{t-1} p(1) \end{aligned}$$

Thus, the model of any word is the set of parameters M={N,p(1),A,B}. To train a recognizer, a library of models must be constructed. For each word, we have to find the number of states, the transition matrix A, the initial-state probability p(1), and the output probability B. At recognition time, the system is given a sequence of observed outputs O where the unknown is identified as that word whose model has the highest probability of forming the observed outputs. For each model Mi, the system determines P{O,M_i}, the probability that M_i gave rise to O, and the unknown is identified as that word j for which P{O|M_j} is the maximum.

Phone models that are HMMs schooled on phonetically-balanced sentences allow for the word pronunciations in a conversation to become move distinct. Each conversation is time-aligned through a hierarchical-grammar speech recognition algorithm that utilizes corresponding conversation, word, and phone models. During this process, a word is characterized by its beginning time and duration. After all words have been characterized, they are combined to reproduce the original conversion. This produces a time-aligned record. Instead of using two single-channel signals, a combined-channel signal has proven to be more efficient in time and error prevention in this time-alignment process. Because empty portions transpire in a signal when a conversation is analyzed, the frequency of errors occurs more in the single-channel signal. Using the combined-channel signals to align the entire conversation requires an effective way of manipulating the simultaneous speech. For example, when two people are engaged in a conversation, words for the first

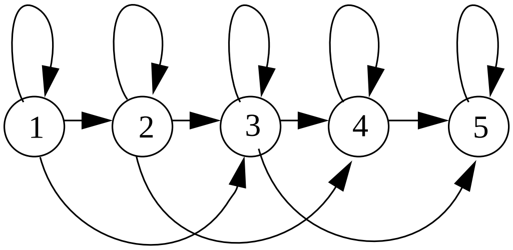

Figure 3. Left-to-Right HMM.

speaker or the second speaker are aligned, but not for both. The recognizer has to decide which of the two paths will align the best and selects that path. One drawback to this approach is that only data from one speaker is realized during simultaneous speech, but, simultaneous speech is normally very brief whenever it occurs. This method does not prove to be extremely apt in enabling the alignment procedure to decipher simultaneous speech.

4. ARTIFICIAL NEURAL NETWORKS

One of the new, emerging fields in computing technology as it applies to speech recognition is in the area of artificial neural networks. Artificial Neural Networks is the youngest and least well understood of the recognition technologies.

The ANN is based on the idea that complex “computing” operations can be implemented by the massive integration of computing components. Each component performs an elementary computation. In these components, memories are stored, computations are performed, and relations formed through patterns of activity of these components.

The ANN contribution to HMMs and the Viterbi search is mainly to serve as an alternative computing structure for carrying out the necessary mathematical operations. The development of more compact and efficient hardware for real-time implementation is a key advantage. The ANN method also enhances future computing tasks by incorporating context or by learning which features are most effective. “Back-end” ANNs can be used to refine the recognition scores and aid in the improvement of performance.

ANN solutions can eventually add massively parallel computing and alternative methods for adaptation to the techniques that speech researchers use. ANNs can adapt and learn which is extremely useful in processing and recognizing speech. ANNs also tend to be more robust and fault tolerant.

5. ALGORITHMS

5.1. Class Structure

A key component in the development of the algorithms is the circular class structure for real time endpoint detections. Virtual functions using the C++ language are used to develop a set of standard calls for opening files, reading parameters and detecting endpoints. This class has sub-classes that are implementations of the different algorithms to be compared. A diagram of the class structure is shown in the figure below. These virtual functions will allow the user to switch between algorithms without recompiling the code. A circular buffer allows for real-time processing.

5.2. Simple Energy

The first endpoint detection algorithm tested is the simple energy calculation. This code is a real-time implementation of the ISIP energy based algorithm.

Parenthesis Equation:

$$y(n) = x(n) - \alpha x(n-1)$$

$$\alpha \rightarrow preemphasis - factor$$

Energy Equation:

$$energy = \left(\frac{1}{N}\left[\sum x^2(n) - \frac{1}{N}[\sum x(n)^2]\right]\right)$$

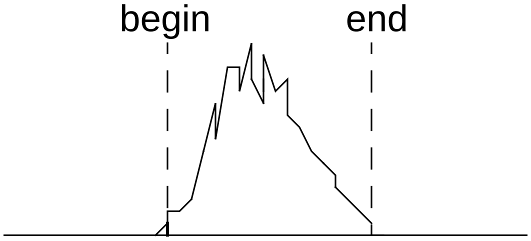

Figure 4. Energy level of the digit “nine”.

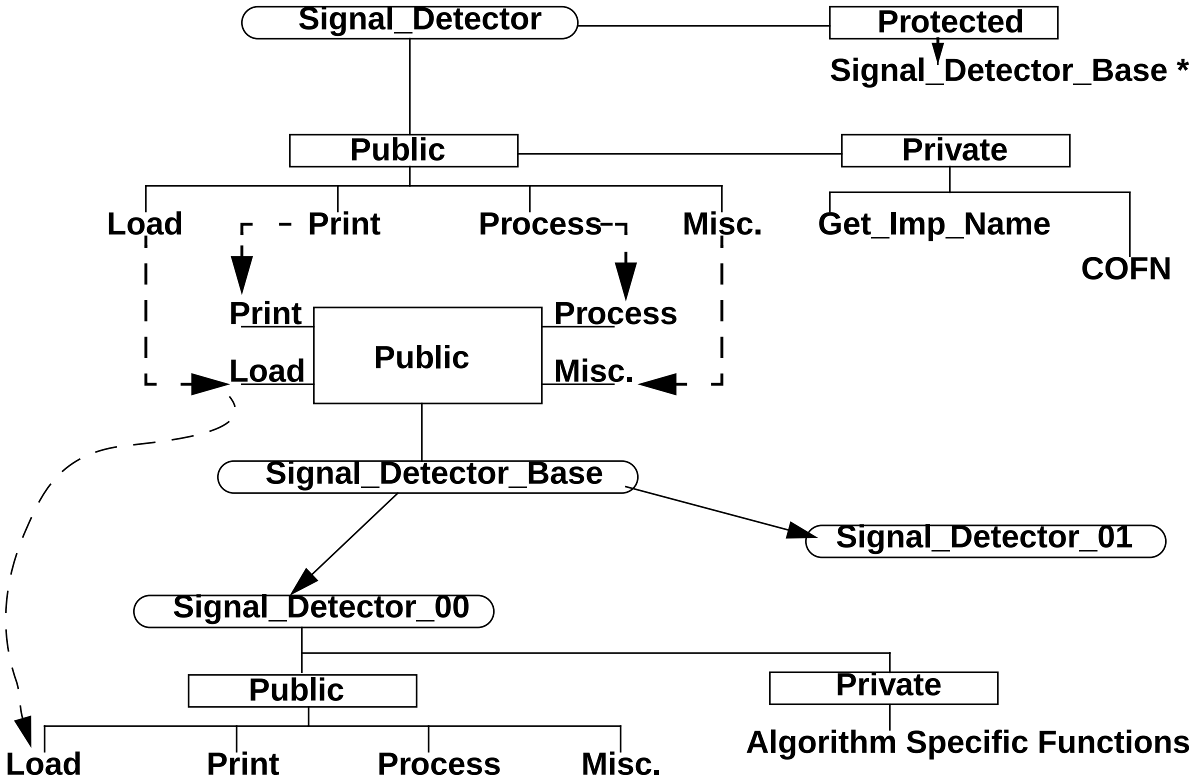

Figure 5. Class Structure Flow Chart.

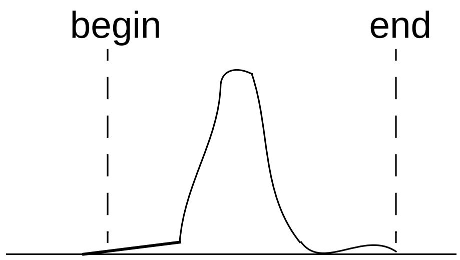

Figure 6. Energy level of the digit “six”.

5.3. Energy and Zero Crossings

The next algorithm developed is using zero crossings. The zero crossing rate is generally considered to be a crude measure of the frequency content of speech. The average zero crossing rate is a good frequency estimate on narrowband signals. However, speech signals are broadband signals. Thus, the zero crossing rate is less accurate. Also, zero crossing rate is not as accurate with a low signal to noise ratio. Since noise and interference will corrupt the signal, it is first smoothed to eliminate multiple zero crossings.

Parenthesis Equation:

$$y(n) = x(n) - \alpha x(n-1)$$

$$\alpha \to preemphasis - factor$$

$$\sum_N [\quad]\mathrm{sgn}[x(n)] - \mathrm{sgn}[x(m-1)]]w(n-m)$$

Zero-Crossing Rate:

$$\mathrm{sgn} = 1 \qquad x(n) \geq pzcthresh$$

$$\mathrm{sgn} = -1 \qquad x(n) \leq nzcthresh$$

$$w(n) = \frac{1}{(2N)} \qquad 0 \leq n \leq N-1$$

$$w(n) = 0 \qquad otherwise$$

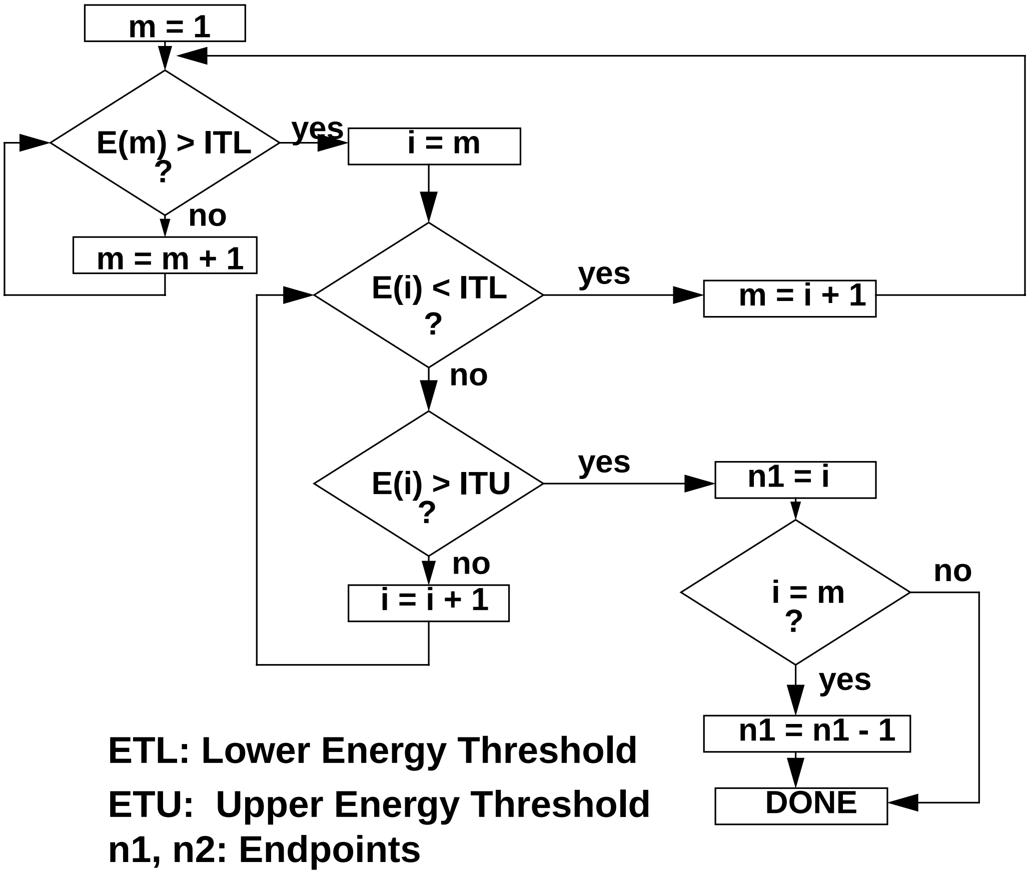

Figure *. Flowchart for Algorithms.

0.1. Public Domain Code (Bruce Lowerre):

The public domain code was written by Bruce Lowerre of Carnegie Mellon University. His C++ based endpointer has evolved from his twenty years experience workig with live input speech signals. The algorithm runs in real time and uses RMS energy and zero crossing calculations to achieve the endpointing. Because the exact start and end of live input is unkown, the algorithm implements a circular buffer. The circular buffer along with the heart of the program, the endpointer, are both implemented as class structures.

The circular buffer has to be large enough to hold the maximum utterance length. The buffer along with the real-time endpointer allows the starting silence before the first utterance to be thrown away. As the start of the utterance is detected in real time, the utterance is buffered until the end is found. Four additional shorts are allocated along with the buffer length in order to make room for the start and end markers. The circular buffer routine assumes that only one module is writing to the buffer and only one module is reading from the buffer. The four data pointers keep track of where data samples are to be written and read.

The read pointer points to the next sample to be read and the write pointer points to the next sample to be written. Neither is allowed to over run the other. Two special pointers, keeper and eod, ae used by the writitng module to mark the start and end of the utterance. In theory, the endpointer will run in real-time. The utterance is read and buffered while the end of the utterance is found in less than real time.

The endpointer class structure determines the start and end of an utterance. It is written to run in real-time with the live input signal. In doing so, the endpointer has to guess to the possible start and end of an utterance. Since these guesses can be incorrect, the endpinter may realize that it has made a mistake. In this cas, special frame labels, EP_RESET and EP_NOTEND, are used to correct a false start and a false end. Frame labels guide the operation of the endpointer. RMS energy calculations and zero crossing counts are used to processs each frame. After the start of the signal is found, the search for the end begins and continues until the parameter, maxipause (end of utterance silence) is exceeded. For non-real time, the search may end for the previous condition or until the end of file is reached. For either case, the EP_ENDOFUTT label is returned , and the last possible guess to the end of the utteracne (EP_MAYBEEND frame) is taken as the end time. When reading a data file, if ever the last EP_MAYBEEND label was cancelled by a EP_NOTEND label, then the endpointer returns “end of utterance not found”. Last, there must be sufficient start silence in order for the endpointer to start processing. This is so the first few frames maybe used to calculate the background noise level.

When either threshold is exceeded for six consecutive frames, the start of utterance is declared. From there, anytime either the energy or the zero crossing count goes below its threshold, that particular frame is labelled as a possible end (EP_MAYBEEND). However, if at anytime after the guess either threshold was exceeded, the last EP_MAYBEEND was cancelled (EP_NOTENDreturned). The parameter, maxipause, defines the maximum amount of time a speaker could pause before the endpointer stopped processing and returned the endpoints. This holds true also for silence found at the end of a speech file.

Bruce Lowerre wrote his endpointer to detect one utterance at a time. The code is maximized for single channel, 8K Hz data. His data files consist of 8 bit, 8K Hz a-law data; 8 bit, 8K Hz mu-law data; and 16 bit linear data which are byte swapped. He indicates that .raw files renamed with a .16l extension would not be byte swapped. Rather than rename the .raw files in the project database, the swap procedure for .raw files was commented out.

The algorithm was not integrated into the class structure with the other alorithms. Instead, the Lowerre code was ran separately with its corresponding parameters set equal to those in the parameter list of the class structure. Because of the nature of the Lowerre code, maxipause was the only parameter equated. In order to compare the Lowerre code with the others, his algorithm was forced to detect multiple utterances. In doing so, maxipause was equivalent to the minimum utterance separation parameter in the parameter file. Recall that when the silence count after the end of an utterance exceeds maxipause, “end of utterance” is declared. The same holds true here. Within a multiple utterance file, when the minumum utterance separation is greater than maxipause, and the silence count exceeds maxipause, the endpointer declares “end of utterance” and writesthe endpoints. Before any modifications, the endpointer’s processing flag would be set to FALSE and processing would cease. However, for multiple detection, the flag is set to TRUE and processing continues in the same manner the first utterance was detected. One can visualize the multiple utterance file being separated into blocks of single utterances. The Lowerre code performs it natural functions on each block.

The energy and zero crossing thresholds were kept constant. They were raised, however, to prevent the code from detecting interword blips. Since the program needed to generate an endpoint file with the starting and ending times of each utterance, the beginning times were equated to the first EP_NOTENDs of each block. The begin time for the very first utterance was equatedto the start frame of the signal. The ending times were equated to the last EP_MAYBEENDs of each block.

6. The Database

The training database for the algorithms was recorded in the ISIP Demo Room under typical office environment conditions (no special care was taken to keep the noise level of the ambient environment low). The data was recorded as two-channel 16 kHz data and stored as 16 bit integers. Two sets of microphones and pre-amps were used in the recording of the data as described in Table 1.

The corpus is composed of three male and three female speakers with unique voice characteristics. Table2 depicts the recording set each speaker used. The use of different recording sets is of importance because one piece of equipment does not have the same set of characteristics such as gain or resolution as the other similar piece of equipment.

Each word spoken in the database was carefully chosen. To fully test the enpoint detection software, words that end on differned sounds, pitches, or frequencies were selected.

SET	(1)	(2)
microphone	Audio Technica AT9100 unidirectional dynamic	Radio Shack 33-2001A omnidirectional dynamic
pre-amp	Teac Cassette Tape Deck	Radio Shack 15-961 Video/Audio Mixer

Table 1. Recording Equipment Sets.

Speaker	Gender	Recording Set
spk_f01	female	AT9100
spk_f02	female	RS 33-2001A
spk_f03	female	RS 33-2001A
spk_m01	male	AT9100
spk_m02	male	AT9100
spk_m03	male	RS 33-2001A

Table 2. Use of Recording Sets.

The filename convention for the data storage is as follows:

spk_f01/spk_f01_05.raw

“f01” denotes female speaker number 1;

“05” denotes utterance number 5;

“.raw” denotes the binary speech data.

Table 3 contains the information that was spoken in each respective utterance type.

Utterance	Type
1	Isolated Digits
2	Teens
3	Multi-syllable Digit Strings
4	Long Digit Strings
5	Phrases
6	Sentences
7	Spontaneous Speech

Table3. Utterance Type.

- Isolated digits spoken are as follows:

“zero”, “one”, “two”, “three”, “four”, “five”,“six”,“seven”,“eight”,“nine”, and "oh"”.

- Teens spoken are as follows:

“thirteen”, “fourteen”, “fifteen”, “sixteen”, “seventeen”, “eighteen”, and “nineteen”.

- Multi-syllabi digit strings spoken are as follows:

“twenty-seven”, “forty-four”, “sixty-five”, “seventy-seven”, and “ninety-nine”.

- Long digit strings spoken are as follows:

“one million two hundred and thirty-six”, “three hundred thousand one hundred and eighty-eight”, and “thirteen million four hundred thousand three hundred and sixty-one”.

- Phrases spoken are as follows:

“flexibility in a job applicant is important”,“the 1996 edition of the Far Side calendar”, “the ghost relinquished her multitude of tricks”, “high definition television”, and “Webster’s tenth annual unabridged dictionary”.

- Sentences spoken are as follows:

“We hold these truths to be self-evident, that all men are created equal, “that among these rights are life, liberty, and the pursuit of happiness", and "it is the right of people to alter or to abolish it, and to institute new government, laying its foundation on such principles and organizing its powers".

● Spontaneous speech spoken was a reply to the statement:

"Please tell me what you like and dislike about your courses/job this semester." Each utterance is approximately five sentences long.

Each speaker spoke the same utterances, but he or she was allowed to pick the order. The total training database consisted of 42 different files.

5.4. Subject Selection

Of course, such evaluations depend heavily on the diligence of the subjects. Subjects with the following characteristics were selected based on *a priori* knowledge of their capabilities:

- American citizens for whom English is their first and primary language;
- normal speech (no known speech impediments or other abnormalities);
- college-educated adults (a mixture of undergraduate students and faculty);
- distinct voice characteristics.

5.5. The Ambient Environment

All attempts were made to provide a normal, office environment ambient environment for the evaluation. The room used during the evaluation was a terminal room with a fairly low background noise level (approx. 50 dBA). This room contained a number of X terminals and small Unix workstations, and was fairly quiet, but during the period of the collection of the database, the room was lightly populated, the door to the room was left open, and hence the room was relatively noisy.(but did contain several subjects performing the experiment as well as an occasional group of students working at the terminals).

5.6. The D/A Audio Interface

The D/A audio interface employed in this project was the SONY Super Bit Mapping, High Density Linear A/D D/A Converter 60ES. This D/A audio interface is a high quality digital audio tape deck.

The individual constructing the database was allowed to adjust gain, but not any other audio parameter, in playing the data.

5.7. Data Collection

To collect the data, the researcher recruited one speaker at a time. Several ISIP tools were employed to record and play the data. Once the command to record the data was issued to the machine, the researcher prompted the speaker to utter the utterances of the desired utterance type. For example, if the female speaker one was asked to speak the isolated digits, the information for those utterances was stored in the file "spk_f01_01.raw". Once the speaker's utterances were recorded, the utterances were checked by playing them and identifying that all utterances for that respective filetype were present. After all seven utterance types for a speaker were recorded and stored in a directory, the data was converted from two-channel data to single-channel data using a simple C program to down-sample the data. The data was converted to single-channel because two-channel data occupies more space on the system and one channel is sufficient to analyze the data.

5.8. Handmarking the Data

Once all data was collected, each data file was handmarked. Each data file was viewed using the ISIP tool plot_signal. Once the utterance was displayed on the screen, the endpoints were marked. To decipher where to precisely mark the endpoints, a significant change in energy was a key factor. Also, a recognizable gap space existed between the utterances which aided in the handmarking of the endpoints. The endpoints for each file were stored in a file with the extension ".endpts". For example,

spk_f01/spk_f01_05.endpts

"f01" denotes female speaker number 1;

"05" denotes utterance number 5;

".endpts" denotes the endpoint file.

In the endpt file, the endpoint information was stored as follows:

```
0 beg_time = 1.705 secs   end_time = 1.915 secs

1 beg_time = 2.533 secs   end_time = 2.763 secs
```

The ‘0’ or the ‘1’ in front of beg_time denotes utterance number.

Figure 4 displays the signal as viewed with plot_signal. Figure 5 shows the handmarked endpoints using the ISIP tool plot_endpoints. Plot_endpoints references both the .raw file and the .endpts file. In Figure 5, accurate endpoints of each utterance are clearly marked. When handmarked incorrectly, plot_endpoints returns Figure 6. After the third utterance in Figure 6, an error has been detected and is easily seen.

6. EVALUATION

An integral part of any software development is evaluation and benchmarking. One of the challenges in benchmarking endpoint detection algorithms is that there are no hard rules for determining the endpoints. Another problem is that the algorithm may not detect the exact number of utterances that are listed in the hand-marked database.

To evaluate the algorithms under development, an objective scoring function was developed. This function determines and records the accuracy of the endpoints. One file per speaker was scored during a single pass of the function. The scoring function compared the handmarked data with the output of the algorithms.

For error difference between the handmarked data and the output of the algorithms in the range (-0.05,0.05) no penalty was assigned. If the difference was between [0.05 ,0.5] or [-0.05,-0.5], a penalty that was linearly scaled was assigned. If the difference in the two values was greater than 0.5 or -0.5, a penalty of 1 was issued. Once all penalties were calculated, they were summed to a grand total. Penalties were further separated as to whether the penalty was an insertion or a deletion. For this application, a deletion denotes that an utterance endpoint was not detected by the algorithm and an insertion implies that the algorithm detected an additional endpoint in the file.

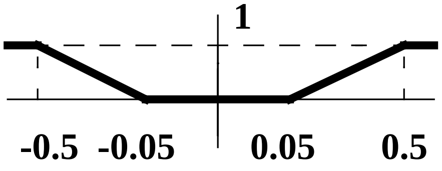

Figure 7. Scoring function.

The important part of evaluation was setting of parameters for the different speech types. The results of the evaluation are biased by these settings. After a lot of iterations we decided of the parameter settings which give us the best performance of all the algorithms.

7. SUMMARY

A major cause of errors in isolated word speech recognition systems is the inaccurate detection of the beginning and ends of speech utterances. It is important in many applications, such as speech recognition over telephone lines, where the SNR is very low, to have a robust endpoint detection algorithm. Such algorithms still have application in a wide variety of areas including speaker dependent speech recognition. Characteristics of human speech such as low energy fricatives and plosives at the end of utterances make endpoint detection difficult.

Many algorithms have been proposed for endpoint detection. Features such as energy [1-3], zero crossing rate [1-4], and periodicity [4], have been used. These algorithms are straightforward to implement when

Algorithm	Bruce Lowerre Code			
Speakers	sub	del	ins	Total
Female01	.284	4	3	7.284
Female02	.231	4	3	7.231
Female03	.151	3	2	5.151
Avg Fem	.222	3.67	2.63	6.562
Male01	.262	6	1	7.262
Male02	.261	5	1	6.261
Male03	.301	10	3	13.30
Avg Male	.274	7	1.67	8.944

Table 6. Bruce Lowerre Code Score.

Algorithm	Simple Energy			
Speakers	sub	del	ins	Total
Female01	.17	1	0	1.17
Female02	.18	0	1	1.18
Female03	.22	1	0	1.22
Avg Fem	.19	2	1	1.19
Male01	.32	1	0	1.32
Male02	.33	0	1	1.33
Male03	.18	0	0	0.18
Avg Male	.28	1	1	0.94
Utt Type				
Isolated Digits	0.11	0	0	0.11
Teens	0.10	2	1	3.10
Multi-Syl Dig	0.12	0	0	0.12
Long Dig	0.17	1	0	1.17
Short Sen	0.24	0	0	0.24
Long Sen	0.20	0	0	0.20
Spon Spc	0.73	0	0	0.73

Table 4. Energy Score.

Algorithm	Energy & Zero Crossing Rate			
Speakers	sub	del	ins	Total
Female01	.16	1	0	1.16
Female02	.19	0	1	1.19
Female03	.21	1	0	1.21
Avg Fem	.19	2	1	1.18
Male01	.31	1	0	1.31
Male02	.34	0	1	1.34
Male03	.18	0	0	0.18
Avg Male	.28	1	1	0.94
Utt Type				
Isolated Digits	0.12	0	0	0.12
Teens	0.06	2	1	3.06
Multi-Syl Dig	0.11	0	0	0.11
Long Dig	0.15	1	0	1.15
Short Sen	0.28	0	0	0.28
Long Sen	0.32	0	0	0.32
Spon Spc	0.71	0	0	0.71

Table 5. Energy and Zero Crossing Score.

compared to the more recently proposed algorithms based on Hidden Markov Models (HMMs) [5] and Neural Networks (NNs) [6]. The latter are more robust and accurate, but are very complex to initialize, and are usually highly channel dependent. In fact, when using NNs for speech recognition, a conventional algorithm performs the initial segmentation of the speech signal, and passes it to the NN for additional processing.

The motivation for developing our algorithm resulted from the need for such an algorithm in a project involving the creation of a large Japanese language speech database. This database contains digits, isolated and four-digit sequences, monosyllables, control words, and several hundred Japanese city names. The data originally were stored on digital audio tapes, and are being automatically processed for use in speech recognition technology development by the endpoint detection algorithms described here.

A comprehensive comparison of the several popular algorithms will be presented. The primary aim of our implementation is to provide the user with an assortment of algorithms from a common software interface. Our implementation makes extensive use of object-oriented concepts, data-driven programming techniques, and a novel virtual class methodology.

We have created a reference database for our project. It is composed of speech data recorded in a moderately noisy environment using a 16 kHz sample frequency. Isolated words, sentences, and paragraphs are used for the prompting text material. The database contains three male speakers and three female speakers. Reference endpoints were created by hand-marking the data using visual and auditory tools. An objective evaluation paradigm was developed to support optimization of each algorithm on the reference database.

The implementation described in this paper is available as public domain software via anonymous ftp at isip.msstate.edu. Supporting tools, such as an endpoint plotter, signal plotter, and signal manipulation programs are also available.

8. RESULTS

The parameters which effect the performance of algorithms the most are the **minimum utterance separation** and the **minimum utterance duration.** The minimum utterance separation is the minimum gap between speech units we expect the algorithm to segment. If we are dealing with short sentences the units of speech are each of the short sentences and the minimum utterance separation will be about 1 sec. which is nominal by any speaking standards. The Minimum utterance duration is the minimum duration of the speech unit we expect the algorithms to segment the input speech data into. For example, if we are dealing with long digit strings the minimum utterance duration ranges from 1 to 2 secs. depending on the application. These settings affect the performance of the algorithms in that , if these parameters are not correctly set there will be more insertions and deletions than what the algorithm could optimally perform.

Type of utterance	Minimum Utterance Separation (secs.)	Minimum Utterance Duration (secs.)
Isolated Digits	.20	.06
Teens	.40	.10
Multi Syllable Digits	.30	.10
Long Digit Strings	1.0	.10
Short Sentences	.90	.10
Long Sentences	.90	.10
Spontaneous Speech	3.0	.10

Tables 4, 5 and 6 show the results of the evaluation of the energy algorithm, the zero crossing & energy algorithm and the public domain code. Some conclusive results can be seen from these tables. The energy and zero crossing algorithms perform equally well in most of the cases. The advantage of using zerocrossings can be seen more clearly when dealing with speech data with a low SNR like that recorded from a telephone. We have done our experiments with data having an SNR in the range of 50dB-60dB. This could be the reason why the performances of the two algorithms was identical. The Zero crossing algorithm does, however, perform well when dealing with spontaneous speech.

The performance analysis results of the Bruce Lowerre code are misleading. The code was originally written to detect endpoints of a single utterance per file. The code was changed so as to detect multiple utterances per file. The parameters for the code are hard coded and hence the performance was bad. Though the score for substitutions was close to that of other algorithms, the number of deletions and additions was far too high. The performance of the algorithms differentiating on the sex of the speakers is identical, though the performance on speech data of females is better than that for male speakers.

9. CONCLUSIONS

In this paper, a comparison of three algorithms for speech endpoint detection has been done. The performance evaluation shows that the energy and zero crossing algorithm performs marginally better than the energy algorithm. The Bruce Lowerre code is not very suitable for processing of multiple-utterance-per-file sort of data. The results encourage us to use the energy and zero crossing algorithm as an automatic word data trimming tool. In the near future we propose to add new algorithms to the existing structure of the code which use features like spectral slope and periodicity. We plan to evaluate the performance of the algorithms using data with low SNR , especially telephone data so that the code can be made the front end of an automatic telephone data collection system. Tests have to performed on data from a larger database than the one we used. The software will be available on public domain after fine refinements to the algorithms are made. The supporting software tools like the signal plotter, the endpoint plotter etc. will also be made available.

12. ACKNOWLEDGEMENTS

The authors are grateful to Dr. Joseph Picone for all the support he has given towards this project and also supplying us with software tools to perform this project. We would also like to thank Dr. William Ebel for his suggestions . We would like to acknowledge the work done by Mr. Paul Kornman towards this project. He was extremely instrumental in giving this project a head start without which the project would not be a reality.

10. REFERENCES

1. L.R. Rabiner and M.R. Sambur, “An algorithm for determining the endpoints of isolated utterances,” *Bell Syst. Tech. J.*, Vol. 54, pp. 297-315, 1975.
2. R. Tucker, “ Voice Activity Detection Using a Perodicity Measure,” *IEE Proceedings,* Part 1, Communications, Speech, Vision, Val. 139, pp. 377-380, Aug. 1992.
3. J. Junqua, B.Mak, and B. Reaves, “A Robust Algorithm for Word Boundary Detection in Presence of Noise,” *IEEE Trans. on Speech and Audio Processing,* Vol. 2, No. 3, pp. 406-412, July 1994.
4. L.F. Lamel, L.R. Rabiner, A.E. Rosenberg, and J.G. Wilpon, “An Improved Endpoint Detector for Isolated Word Recognition,” *IEEE Trans. Acoustics., Speech, Signal Processing,* Vol. ASSP-29, pp. 777-785, Aug. 1981.
5. B.Wheatley, et. al.,“Robust Automatic Time Align men of Orthographic Transcriptions with Unconstrained Speech, *Proc. ICASSP 92,* Vol. 1, pp. 533-536, 1992.
6. D.P. Morgan and C.L. Scaffold, *Neural Networks and Speech Processing,* Keller Academic Publish errs, Narrowly, Massachusetts, U.S.A., 1991.D.P. Morgan and C.L. Scaffold, Neural Networks and Speech Processing, Keller Academic Publishers, Narrowly, Massachusetts, U.S.A., 1991.
7. Noboru Sugamura ,”Continuous Speech Recognition Using Large Vocabulary Word Spotting and CV Syllable Spotting,” *Proc. ICASSP 90,* vol. 1 pp. 121-124.
8. Robert J. McAulay and Thomas F. Quatieri,”Pitch Estimation and Voicing Detection based on a Sinusoidal Speech Model,” *Proc. ICASSP 90,* vol.1, pp.249-252.
9. J.G.Wilpon, D.M.Demarco,R.P.Mikkilineni,”Isolated Word Recognition Over the DDD Telephone Network - Results of Two Extensive Field Studies,” *Proc. ICASSP 88,* vol. 1 ,pp.55-58.
10. P. Haffner, M. Franzini and A.Waibel,” Integrating Time Allignment and Neural Networks for High Performance Continuous Speech Recognition,” *Proc. ICASSP 91,* vol.1 ,pp. 105 -108.
11. P. Ramesh, S.Katagiri and C-H lee,” A New Connected Word Recognition Algorithm Based on HMM/LVQ Segmentation and LVQ Classification,” *Proc. ICASSP 91,*vol. 1 ,pp. 113-116.
12. A.Ljolje and M.D.Riley,”Automatic Segmentation and Labelling of Speech,” *Proc. ICASSP 91* ,vol. 1 ,pp. 473-476.
13. Ronald P. Cohn,” Robust Voiced/Unvoiced Speech Classification Using a Neural Networks,” Proc. *ICASSP 91* ,vol. 1,pp. 437 -440.
14. C.Tsao and R.M. Gray,” An endpointer fr 1pc speech using residual error look-ahead for vector quantization applications,” *Proc. ICASSP 84*, pp. 18b.7.1 - 18b.7.1.4.
15. H.Ney,” An optimization algorithm for determining the endpoints of isolated utterances,” *Proc. ICASSP* 81,vol .2, pp. 720-723.
16. J.G.Wilpon and L.R.Rabiner,”Applications of Hidden Markov Models to Automatic Speech Endpoint Detection,” *Computer Speech,Language*,vol. 2, pp. 321 -341,1987.
17. B. Reaves,” Comments on an improved endpoint detector for isolated word recognition,” *Corresp. IEEE Acoust.,Speech, Signal Processing,* vol.39.pp. 526-527, Feb.1991.
18. G.R.Doddington, J.Picone, and J. Godfrey,” The LPC Trace as an HMM Development Tool,” *Journal of the Acoustical Society of America*,vol.84,suppliment 1,Fall 1988.
19. J. Hansen and O. Bria,” Lambard effect compensation for robust automatic speech recognition in

noise,” *Proc. ICLSP 90,* pp. 1125 -1128.

20. L.R. Rabiner and S.E. Levinson,” Isolated and connected word recognition: Theory and Selected Applications,” *IEEE Trans. Communications*,vol.29,pp. 621-659.

21. L.R. Rabiner, “ On creting reference templates for speaker independent recognition of isolated words,” *IEEE Trans. on Acoustics, Speech and Signal Processing,* vol. 26, pp. 34 -42, Feb. 1978.

22. L.R. Rabiner,” A tutorial on hidden Markov models and selected applications min speech recognition,” *Proceedings of the IEEE,* vol.77,pp.257-285, Feb. 1989.

23.T. Svendsen and F. Soong, “ On the Automatic Segmentation of Speech Signals,” *Proc. ICASSP 87,* vol. 1 , pp. 77-80.

24. H.C. Leung, V.W. Zue,” A procedure for automatic allignment of phonetic transcriptions with continuous speech,” *Proc. ICASSP 84,* vol. 1 ,pp. 2.7.1 - 2.7.4.

25. B.S.Atal,” Efficient coding of LPC-parameters by temporal decomposition,” *Proc. ICASSP 83,* vol. 1 ,pp. 81 -84.

26. A.E. Rosenberg, F.K. Soong,” On the use of instantaneous and transitional spectral information in speaker recognition,” *Proc. ICASSP 86,* vol.2 , pp. 17.5.1 - 17.5.4.

27. J.G. Wilpon, L.R. Rabiner,A. Bergh,” Speaker independent isolated word recognition using an airline vocabulary,” *Journal of the Acoustical Society of America,*” vol. 72, pp. 390-396, Aug. 1982.

28. Douglas O’ Shaughnessy, *Speech Communication-Human and Machine,* Reading Mass.,Addison-Wesley,1987.

29. A.V. Oppenheim and R.W. Schafer, *Discrete Time Signal Processing,* Englewood Cliffs, N.J.: Prentice Hall, 1989.

30. J.G. Proakis et.al., *Discrete Time Processing of Speech Signals*,Englewood Cliffs,N.J., Macmillan,1993.

31. L.R. Rabiner,” On the application of energy contours to the recognition of connected word sequences,” *AT&T Bell Labs Tech. Journal 63*,pp. 1981-1995, 1984.

32. J.G.Wilpon, L.R. Rabiner and T. Martin,” An improved word detection algorithm for telephone quality speech incorporating both syntactic and semantic constraints.,” *AT&T Bell Labs Tech. Journal 63*, 479-497, 1984.

33. Thomas Parsons, *Voice and Speech Processing*,N.Y,. McGraw-Hill , 1987.

34. A. Nejat Ince, *Digital Speech Processing - Speech Coding , Synthesis and Recognition*,Norwell,Mass., Kluwer Academic, 1992.

35. M. Savoji,” A robust algorithm for accurate endpointing of speech,” Speech Communication, (8):45-60,1989.

36. P. Mermelstein,” Automatic segmentation of speech into syllabic units,” Journal of Acoustical Society of America,vol. 58,pp. 880-883,1975.

37. J.P. van Hemert, “ Automatic Diphone Preparation,” IPO Annual Progress Report, 1985,pp. 656-659, June 1985.