A SPEECH-RECOGNITION-BASED SYSTEM FOR CONVERTING HEBREW TEXT TO PHONETIC REPRESENTATION

Gal Ben-David * , David Malah ** , Uzi Ornan ***

Abstract

A system for converting Hebrew text into phonetic text is presented. The system applies speech recognition techniques for identifying vowels in spoken text and inserts these vowels in the original vowel-less Hebrew text. The original Hebrew text is presented to the user on his computer terminal. The user then reads the text into a microphone and the system analyzes the input voice and converts each word into a string of Linear Prediction coefficients (LPC) vectors which are then compared to vectors stored in a set of codebooks (one codebook per vowel) to find the best match. The identified vowels and their location are then input to a Dynamic-Time-Warping algorithm which matches the acoustic information with the text in order to correctly position the vowels in each word using Hebrew syntax rules. The system could be used in preparing text for automatic Hebrew Text-To-Speech (TTS) systems and in computer aided linguistic studies.

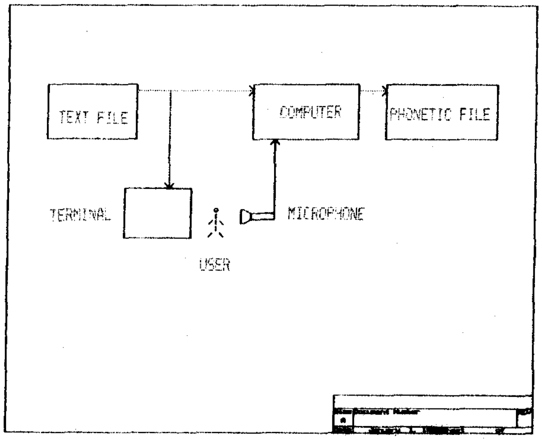
I. INTODUCTION

Ordinary Hebrew text is characterized by a concise representation in which vowels are not explicitly written . Vowel signs do exist (in the form of dots and under-bars , etc. - known as 'Nikud') but since they are generally omitted , the reader must insert by himself the missing vowels However , different vowel insertions into a given word may result in different meanings. Hence , the reader must understand the context in which a word is given in order to correctly pronounce it . For example the word "הרכבת" can be interpreted with no 'Nikud' as having one of the following meanings : 'the train' , 'you assembled', 'you gave a ride' , 'vaccination'.

Another problem in the word is that the "ה" letter may not be a part of the word but a similar to the English 'THE'.

This has been a major obstacle in developing automatic TTS systems in Hebrew. Of course , if the text includes the vowels, the problem would not be much different than in English. However , manual insertion of vowels with a word processor (There are only five vowels in Hebrew : a,e,i,o,u and only one consecutive vowel "ei") is slow and inconvenient.

The approach proposed in this work is to semi - automatically convert ordinary (vowel - less) text into a phonetic representation by letting a human operator read the text from a terminal screen into a microphone and then have the computer identify the vowels in each word and insert them in the correct positions and generate a phonetic file.
The building blocks of the proposed system are shown in Fig. 1.



rig. 1 : proposed system building blocks

* Gal Ben-David is a graduate student at the Technion , and has done this work in a graduate lab-course at the Signal Processing Laboratory of the E. E. deptartment ,under the supervision of the other authors.
**David Malah is presently on sabbatical at AT&T Bell Labs, Murray Hill, NJ.
*** Uzi Ornan is presently with the Computer Science Dept., Technion.

Electrical Engineering Dept.
Computer Science Dept.
Technion -
Israel Institute of Technology
Technion City , Haifa 32000 , Israel

The system makes use of well-known speech recognition algorithms but applies them to this new application . The main effort in this work was directed to the following :

1. Generating speaker dependent vowels-dictionaries.
2. Detecting word boundaries and identifying vowels in isolated words.
3. Inserting the identified vowels in the correct positions in the given text.
4. Building dedicated hardware for speech I/O and writing a user friendly operating system for the given task.

II. SPEECH PARAMETERS

The system displays few lines of text from the file to be converted, and the user is prompted to read the text into a microphone (in an isolated-word fashion). The input speech filtered with a 300-3200 Hz Band-Pass-Filter and sampled at 8KHz, using 12-bit linear A/D converter Word boundaries are then detected using energy envelope information and zero-crossing rate [1] . After displaying the text the system waits for a "start of word" declaration. A "start of word" is declared when one of the following conditions is satisfied within a speech frame (N=256 samples):

1. Energy estimation (1) passed a given threshold. This condition is typical of a voiced beginning of a word.

$$(1) \qquad E(n)=\sum_{j=n-N}^{n} x^2(j)$$

2. Zero - crossing count passed another threshold, a zero-cross is declared whenever the signal passed pre - determined positive and negative levels. A word with an unvoiced beginning is characterized by this condition.

A word decay is recognized by a decreasing of the energy and zero-crossing level below previous thresholds for a period of 256 mS. This delay prevents an "end of word" declaration while in-word silence period . After "end of word" declaration , system search backward in a symmetric manner for exact word decay position.

A common model for a speech signal is an All-pole model named Linear Prediction Coding (LPC) [3] .The LPC model is shown in Fig. 2

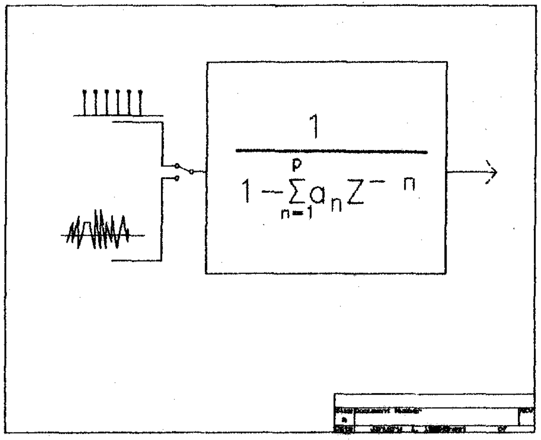


Fig. 2 A LPC model

A white Gaussian noise or impulse train are input to an All-pole filter , while the "a" coefficients represent the vocal information.
The LPC analysis is performed on Hamming windowed speech frames of 16mS, and the LPC coefficients extracted are used to represent each frame of speech.
The resulting string of LPC vectors obtained for each word is used to identify the vowels in each word by comparing each LPC vector to the vectors in the vowels dictionaries.

A distance measure for non-similarity between two LPC "a" coefficients vectors is needed. The well-known Itakura-Saito distance measure [2] is used.

III. VOWEL DICTIONARIES

The dictionaries were prepared in a training phase of the system's operation, in which 75 different Hebrew words were spoken by a single user and were sampled and analyzed by the system . Then from apriori knowledge of the acoustic input,130 LPC vectors were manually selected to represent each vowel.These vectors provides then the training sequence for constructing a representative smaller dictionary for each vowel . An iterative clustering algorithm known as the LBG algorithm [4] (commonly used in Vector -Quantization applications) was used . The LBG algorithm is an iterative process for dividing the training sequence into pre-determined number of sets while minimizing the mean squared error of the input vector and dictionary vectors generated by the algorithm . Following this clustering process , a total of 60 vectors(representing the five vowels) were obtained . The size of the overall dictionary was further reduced , manually , to improve its performance for the given task, to a total of 49 vectors for the five vowels (not necessarily same size) .

IV. VOWEL RECOGNITION

The final steps in the system's operation are identifying the correct vowel from the vowel candidates mentioned above and inserting the identified vowels in the input text string . The identification of a vowel is based on weighting the data obtained in several consecutive frames which are within a stationary segment of the word , and comparing the weighted distance to a threshold . If the threshold is passed a "No-vowels" decision is made.

More specifically distance is checked for every dictionary vector . Best dictionary vector representing a vowel gets a score c1 second best gets c2 and so on. The decision is made for the vowel with maximum sum of c's. This method relaxes the hard decision of best neighbor and is found to fix some singular error events.Best vowel's distance is compared with a threshold , and bad candidates are neglected.

The boundaries of a stationary segment are found by differencing the distances between consecutive frames to detect acoustic transitions within the word.

The insertion of detected vowel in the text is based on matching the acoustic information and the given text using Dynamic-Time-Warping (DTW) algorithm [5] (commonly used in speech recognition to align two words) , and few Hebrew syntax rules. The Hebrew information is aligned along one axis and the acoustic information along the other axis . The DTW algorithm is a dynamic programming technique , for minimizing a cost function along the route between the edges of the Hebrew and acoustic word.The cost function is based upon weighting good matches like: parallel transitions, Hebrew letters representing good probability for a vowel as shown in the following pairs 'א' and the vowel a , 'ה' and a , 'ו'- o or u, 'י' and i.

V. CONCLUSION

The system was implemented on an IBM-PC by adding special speech I/O,it was tested with an Hebrew song. typical output of the system is as follows :

חו O פי I ם - ה E ם - ל I פ ע A מ י I ם
גע A גו U עי I ם ל E ג A ח A ל .

The system performs with 75% of good recognition , 20% of "not-a-vowel" error while a vowel exists, and 5% wrong vowel recognition. The system presents few future research areas :

1. Searching for other dictionaries for better recognition, performance and automatic preparation of a dictionary after the training procedure by a new user.
2. Using Hebrew syntax rules for more reliable decisions.
3. Implementing the system on a faster DSP chip for accelerating the recognition process , which takes about 10 seconds per word on an IBM-PC.

References

[1] L.R. RABINER , M.R. SAMBUR
"An algorithm for determining the endpoints of isolated utterances "
The Bell System Technical Journal
Vol.54 , No. 2 February 1975 pp 297-314

[2] A.H. GRAY Jr., J.D. MARKEL
" Distance measures for speech processing"
IEEE trans. ASSP Vol. 24 ,No. 5 ,
October 1976 , pp 380-391 .

[3] L.R. RABINER , R.W. SCHAFER
" Digital processing of speech signals"
CH. 8 Prentice Hall 1978

[4] Y.LINDE , A. BUZO , R.M. GRAY
"An algorithm for Vector Quantizer design"
I.E.E.E. transaction on communication
Vol. 28, No. 1 January 1980 pp 84-95

[5] C. MYERS , L.R. RABINER ,
A.E. ROSENBERG " Performance trade-offs in Dynamic Time Warping algorithms for isolated word recognition "
IEEE tran. ASSP Vol. 28 ,
No. 6 December 1980 pp 623-635