

Analysis of Algorithms for Phrase Recognition

J S Mirza¹, Muhammad Umair², S A Hayat³, Asif Hussain⁴

1,2,3: Department of Computer Science
COMSATS Institute of Information Technology, Lahore, Pakistan

4: Department of Physics
Government College University, Lahore

jsmirza@ciitlahore.edu.pk, muhammadumair@ciitlahore.edu.pk
shaukat.ali@ciitlahore.edu.pk, asifphy@hotmail.com

Abstract

Three algorithms, designed to recognize vowels, are analyzed for their complexity class or growth pattern. First a databank was prepared from Vowel loops of Peterson and Barney [1] which was used for vowel recognition. The above three algorithms produced the same vowel-recognition rate of more than 78% on a given short phrase. This paper submits the result of our analysis of the three algorithms to investigate their growth pattern and decide which of the three is the fastest in making decision of vowel recognition. The simple system of vowel recognition, presented in this paper, which leads to phrasal recognition, can serve the purpose of automatic recognition of short phrases in a long wire-tapped message. The phrasal recognition can be advantageous to do a preliminary study whether or not a wire-tapped long message contained phrases of interest. If it does the intelligence agent can get primed to scrutinize the entire message. This scheme can spare a lot of time of heavily loaded intelligence agent deputed to do wire-tapping.

1. Introduction

Phrasal recognition has quite an importance for intelligence agents who wiretap suspects' phone lines to get clues from their conversations what they are up to. These clues can be helpful to prevent hazardous action to happen or convict a suspect if his wiretapped message contains enough information to do so. A project has been undertaken to devise a simple system which would enable an agent to determine whether or not phrase(s) of interest are present in the

wire-tapped recordings which usually are long. Normally the recorded conversations are long and plenty in quantity and agents get bored listening to them. The project was divided into two parts: the first part concerned preparation and testing of algorithm(s) which could determine the presence of a given phrase in a long text; the second part was meant to automate the system, subsequent to the success of the first part, so that the system will search **on its own** the given phrase and alert the agent to minutely go through the conversation in its entirety. The recognition of phrase was simply done through *manual* recognition of vowels of the phrase ignoring the consonants. A software "Pratt" extracted the first two formants of the vowels which were then manually supplied to the algorithm(s) to do the matching with vowel loops.

Three algorithms were designed and tested for their efficacy [2,3]. Having achieved a reasonable success in the desired functionality of the algorithms the second phase of the project was undertaken to work on the algorithms' complexity class so that the best of the three could be employed to possibly work in real time basis. The automatic recognition of phrases of interest from the long wire-tapped messages can quickly provide cue that the message needs to be critically listened to in its entirety. The entire project cropped up in a conversation with intelligence agent in Pakistan who counted the benefits that

can accrue from such a research. The three algorithms are discussed below as well as their growth pattern to decide which of the three can yield the requisite results of recognition quickly. Scheme of further development for automation is described in the section of conclusion.

2. Analysis of Algorithms

Algorithm-1 and algorithm-2 [2,3] used the values of first two formants F1 and F2 of the vowels spoken in the phrase. The values of F1 and F2 known as first and second formants of the vowels were derived by the software “Pratt” from the acoustic analysis of the phrase. In place of Pratt any other software, which does the frequency analysis of speech could be used for this purpose. These two algorithms matched the values of (F1,F2) with vowel loops [1] and determined the vowel symbol corresponding to each vowel. Algorithm-1 is given in Fig.2 along with its analysis. Algorithm-2 differs slightly from algorithm-1 but its analysis yields same result as of algorithm-1.

A Short Urdu phrase “Usama Bin Laden” spoken by an Urdu speaker was used for recognition. For convenience and due to lack of viable data regarding Urdu vowel loops it was assumed that English vowel-loops [1] and their counterparts in Urdu were exactly the same; even though our initial investigation did indicate minor differences. The differences, however, turned out to be minor and therefore negligible for the specific vowel set we were interested in for our testing. An experiment is currently underway at our facility to determine vowel loops of Urdu. A very large number of speakers, about 100 of them, have been selected for this purpose.

The vowel loops were encased inside a grid for matching purpose (grid not shown in the **Figure 1**). The grid comprises a set of parallel vertical lines emanating from F1-axis spaced 25 Hz apart for algorithm-1 and a set of horizontal lines emanating from F2-axis for algorithm-2, spaced 50Hz apart. For a given point (F1,F2) calculated by Pratt the nearest vertical line for F1 is chosen in algorithm-1, and nearest horizontal line for F2 is chosen

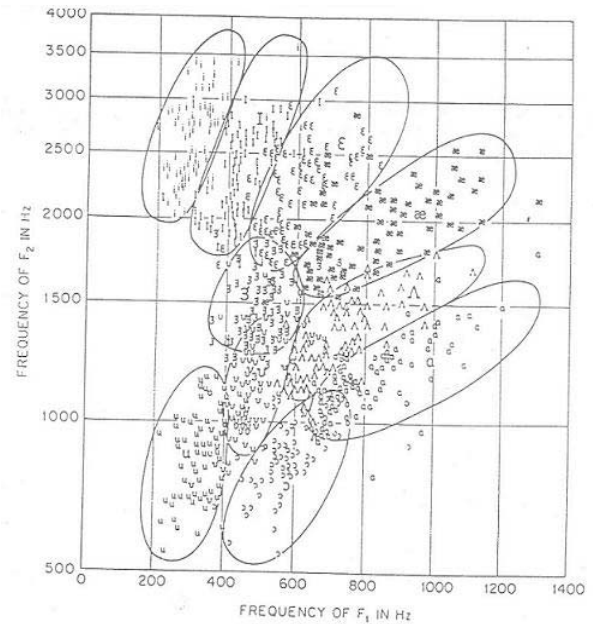


Figure 1: Plot of second formant frequency versus first formant frequency for vowels by a wide range of speakers. (After Peterson and Barney)

in algorithm-2. Each loop in **Figure 1** is represented by a number of vertical lines passing through it; and part of each vertical line inside the loop is characterized by a range of F2 values. For instance for F1=1000 Hz, if the F2 lies in the range of 1120-1480 Hz, then the point (F1,F2) will correspond to the vowel “A”, so on and so forth. Similar explanation holds for algorithm-2; it is determined that for given F2 which loop F1 will lie in.

For a given F1, there can be a number of F2 ranges; one range for each loop. Algorithm-1 determines which loop F2 lies in for a given value of F1, while algorithm-2 determines which loop F1 lies in for a given value of F2. For instance as shown in **Figure 1** for a given F2 of 2500 Hz, the ranges for F1 for “IY”, “I” and “E” are 155-350 Hz, 340-535 Hz and 520-860 Hz respectively. The difference between algorithm-1 and algorithm-2 may be in the consumption of time taken for a decision by the algorithms.

Algorithm-3 uses a table-- not shown here -- with two axes F1 and F2. The grid imposed on vowel-loops yields points of intersection between horizontal and vertical lines which are fed in the table. Each point of intersection would correspond to a certain vowel / vowel symbol. For a given point (F1,F2) algorithm-3 searches the table for the corresponding vowel symbol. Obviously all the three algorithms would yield the same

vowel symbol. However, their speed to reach the result will differ depending upon the values of (F1, F2).

The **Figures 2 and 3** represent the analysis of the algorithm-1 and algorithm-3 respectively. Algorithm-1 and algorithm-2 turned out to be asymptotically quadratic in worst case, while algorithm-3 is asymptotically constant, indicating superiority of algorithm-3 over the other two. One of the disadvantages of algorithm-3 is that it uses more memory compared to others: The F1-axis of loops, scaled by 25 Hz has $1400/25 = 56$ steps, and F2-axis at the scale of 50 Hz has $(4000-500)/25 = 140$ steps. Thus a grid for algorithm-3 requires $56 \times 140 = 7840$ elements to store the entire loops data. The memory requirement however for algorithm-1 and algorithm-2 is much smaller than algorithm-3. For algorithm-1 the number of ranges compared are 221 and for algorithm-2, 218. Thus the memory requirement for algorithm-1 and algorithm-2 are more or less the same. The computational demand will depend upon which vowels are frequently used in the phrasal recognition. Assuming that all the vowels in the phrase have more or less equal chance of being used, then algorithm-1 and algorithm-2 will have nearly same computational time.

3. Results

The three algorithms were manually tested on a short phrase "Usama Bin Laden" spoken in isolation. Three male speakers who had clarity in their speech were selected. They uttered 10 times each a phrase "Usama Bin Laden. The utterances were spoken in front of a microphone connected to the computer directly in a normal room environment. The phrase "Usama Bin Laden" contains a sequence of following vowels (OO, A, A, I, A, UH). The symbols used are typewritten symbols of the vowels and not those of phonetic association. The three algorithms yielded the same recognition rate which was obvious. The growth patterns of the algorithms turned out that algorithm-3 was the quickest compared to other two, and therefore the only candidate in our case for automatic and real time recognition.

4. Conclusion

Algorithm-3, which searches a table to determine vowel symbol for given (F1,F2) is better than algorithm-1 and algorithm-2 in the sense that its complexity class is constant while for algorithm-1 and algorithm-2 is quadratic in worst case and linear in best case. The comparative disadvantage of algorithm-3 is in its requirement of a large memory to store the table and effort to store a large number of table points. Cheap memory offsets the first disadvantage and the second one is obviated because it takes just one effort to fill the requisite table. Complexity class of algorithm-1 and algorithm-2 being same the two will exert same computational demand. Algorithm-3 is definitely superior as far as its growth pattern is concerned.

The values of F1 and F2 for vowel recognition were provided manually to algorithms, which is quite a laborious job. In order to automate the system a software interface needs to be designed so that Pratt's calculated values of F1, F2 could be automatically provided to algorithm3. Only the central parts of vowels were used for recognition, which ought to be so, because the central parts are stable while the beginning and end parts, otherwise known as transient regions of vowels, are heavily affected by the immediately previous and subsequent phonemes. On the average Pratt yielded about 20 values of (F1, F2) for each vowel. Therefore, automation for matching is utterly important. Further development on phrasal recognition in terms of text independent speaker recognition can also be very useful for convicting an accused.

Algorithm-1	Cost	Times
01-While $i \leq n$ && temp == 0	C1	n
02- do while $j \leq m$ && temp == 0	C2	$\sum_{m=0}^n m$
03- if $F_2 \geq a[i][j].min$ && $F_2 \leq a[i][j].max$ temp $\leftarrow 1$; else j = j+1;	C3	$\sum_{m=0}^n (m-1)$
04- i = i + 500	C4	n-1
05-if temp == 1		
06- print a[i][j].symbol;	C5	n-1
07-else		
08- Message "Value not in Valid Range"		

Worst Case Analysis

$$\begin{aligned}
T(n) &= C1 n + C2 \sum_{m=0}^n m + C3 \sum_{m=0}^n (m-1) + C4 (n-1) + C5 (n-1) \\
&= C1 n + C2 (n(n+1)/2) + C3 (n(n+1)/2) - C3 n + C4 (n-1) + C5 (n-1) \\
&= (C2/2 + C3/2) n^2 + (C1 + C2/2 + C3/2 - C3 + C4 + C5) n + (-C4 - C5)
\end{aligned}$$

We can rewrite this expression as

$$T(n) = a n^2 + b n + c$$

This shows the complexity class of the function will be quadratic. Hence asymptotically

$$T(n) = O(n^2)$$

Best Case Analysis

In this situation line 2 will be executed only once i.e. the required frequency F2 is in the first stored range. Thus the cost of the function will be as:

$$\begin{aligned}
T(n) &= C1 n + C2 (n-1) + C3(1) + C4 (n-1) + C5 (n-1) \\
&= (C1 + C2 + C4 + C5) n + (-C2 + C3 - C4 - C5)
\end{aligned}$$

We can rewrite this expression as

$$T(n) = a n + b$$

This shows the complexity class of the function will be linear. Hence asymptotically

$$T(n) = O(n)$$

Fig 2: Algorithm-1: It uses F1 to determine in which vowel loop the point (F1,F2) lies

Algorithm-3

<i>ReadData ()</i>	<i>Cost</i>	<i>Times</i>
1. Initialize the “data” with “space”	C1	1
2. Message “Option to change default interval”	C2	1
3. Read “Option”	C3	1
4. If “Option is “Yes” then		
5. Read “intervalF1”	C4	1
6. Read “intervalF2”		
7. For(i=0;i<= n;i+=intervalF1)	C5	n + 1
8. For(j=0;j<= m;j+=intervalF2)	C6	$\sum_{m=0}^n (m + 1)$
9. Read data[i][j]	C7	$\sum_{m=0}^n (m)$
10. Stop	C8	1

Analysis:

$$T(n) = C1(1) + C2(1) + C3(1) + C4(1) + C5(n+1) + C6 \sum_{m=0}^n (m+1) + C7 \sum_{m=0}^n (m) + C8$$

$$= C1 + C2 + C3 + C4 + C5(n-1) + C6(n(n+1)/2 + n) + C7(n(n+1)/2) + C8$$

$$= (C6/2 + C7/2)n^2 + (C5 + C6/2 + C6 + C7/2)n + (C1 + C2 + C3 + C4 + C5 + C8)$$

$$T(n) = a n^2 + b n + c$$

This is a quadratic equation and hence complexity class will be (n^2). Hence asymptotically

$$T(n) = O(n^2)$$

Fig 4: Data-read part of algorithm-3

<i>FindSymbol()</i>	<i>Cost</i>	<i>Times</i>
1. Read F1, F2	C1	1
2. Symbol = data [F1][F2]	C2	1
3. Display “Symbol”	C3	1
4. Stop	C4	1

Analysis:

$$T(n) = C1(1) + C2(1) + C3(1) + C4(1)$$

$$= (C1 + C2 + C3 + C4)(1)$$

$$= a(1)$$

This clearly shows that its complexity class is constant. Hence asymptotically

$$T(n) = O(1)$$

Fig 3: Find-symbol-part of algorithm-3; it searches table to find the symbol

A relevant research on text independent speaker recognition using source based features is a good source to include speaker recognition [4,5].

Certain values of (F1,F2) yielded wrong vowel symbols. This is because there is some overlapping between some vowels. Evidently such a scenario has to be

catered for. If we do so, the calculated vowel sequence for a given phrase could be more than one. Another algorithm needs to be prepared on the basis of Hidden Markov Model (HMM) which will use the various probabilities of vowel recognition to arrive at the most probable vowel sequence[6].

References

- [1] Peterson, G.E., and Barney, H.L., "Control Methods Used in Study of the Vowels", J. Acoust. Soc. Am., Vol. 24, No.2, pp175-184, March 1952.
- [2] Mirza J.S and Hayat S. A., Wire-tapped Intelligence; Machine Recognition of Specific Phrases to Nab A Suspect. Accepted for publications in the 2005 International Conference on Software Engineering Research and Practice (SERP'05 JUNE 27-30, 2005, Las Vegas, USA)
- [3] Mirza J S, Muhammad Umair , Optimizing Algorithms for Phrase Recognition, 1st CIIT Workshop on Research in Computing (CWRC Spring '2005), Abbotabad, Pakistan, April 2005. (proceedings in print)
- [4] Minh N. Do, " An Automatic Speaker Recognition System" Digital Signal Processing Mini-Project, Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2003.
- [5] Wilermoth B R, M.Phil Thesis: Text-independent Speaker Recognition Using Source-based Features, Griffith University, Australia, January 2001.
- [6] Becchetti C, and Ricotti L P, Speech Recognition: Theory and C++ Implementation, John Wiley and Sons, 2004 (ISBN: 9812-53-107-6)