Optimal Design of a Bengali Virtual Keyboard seeking improved user perception for a better text entry rate

(*Human Computer Interaction*)

Mr. Panthadeep Bhattacharjee
Assistant Professor, School of Computer Engineering
KIIT University
Bhubaneswar, India
panthadeep@gmail.com

Mr. Lalit Kumar Vashishtha
Assistant Professor, School of Computer Engineering
KIIT University
Bhubaneswar, India
lalitkvashishtha@gmail.com

***Abstract*—In this paper an approach is proposed to derive an optimal layout of a Bengali Virtual Keyboard and hence forth seeking the desirable placement of all the keys in that layout within the constraints of space and the key size, thereby seeking to improve the text entry rate for an individual with normal human perception abilities. The approach is partly segmented in identifying the initial layout to work with, obtained through a design space exploration technique, which essentially forms an input to the Optimal Letter Arrangement algorithm. The algorithm involves a pool of keyboards to start with for each of which the theoretical text entry rate is evaluated based on Fitts' hand movement law as the fitness function. The fit keyboards make a hop to the next generation after which they go through a randomization phase in letter arrangement and subsequent fitness evaluation. The unfit keyboards inherit a known layout and make the subsequent hop. A keyboard stops the procedure the moment its text entry rate becomes stagnant and pulls itself out of the pool. In the end the pool becomes empty and the keyboard having the highest text entry rate from the removed set of keyboards is chosen after which it is subjected to practical usage by the users.**

***Keywords*—Virtual keyboard, Fitts' movement law, Bengali, optimal, layout.**

I. Introduction

A Virtual Keyboard is a software component that allows a user to enter characters. A Virtual Keyboard can usually be operated with multiple input devices, which may include a touch screen, an actual keyboard and a computer mouse. The available virtual keyboards in Bengali in general do not show improved performance in terms of the text entry rate. Hence the importance of producing a virtual keyboard in Bengali with an improved text entry rate along with the user friendly layout has been of utmost need. The prominent existing keyboards in Bengali language like the *AVRO* keyboard and also the *AVRO* online keyboard do not show an encouraging text entry rate in spite of the fact that it has been one of the important applications that has been used to type Bengali letters. Another important application that exists is the *Virtual Bangla Keyboard* which has its layout chalked on the QWERTY layout of English language but it has its own limitations too. The application provided by Microsoft is phonetics based. The same disadvantage exists with the Google translator too with the application being phonetics based which requires the typing of the English words like ”ka” “kha”. These limitations would be a hindrance in meeting the rampaging text entry scenario.

With the target language as Bengali, the design principles applied in the designing of an English keyboard cannot be applied with that of a Bengali version. This can be implied from the fact that in the Bengali script its vowel graphemes are realized not as independent letters but as diacritics attached to its consonant letters. It is written from left to right and lacks distinct letter cases. It is recognizable by a distinctive horizontal line running along the top of the letters that links them together. The number of characters in the layout is evidently more than it is for an English layout. With the frequent use of conjuncts, a typical Bengali text, a typical Bengali text requires the usage of a particular set of characters more as compared to the rest. There are equivalent vowel-marks as well as adjuncts which have significant involvement in the writing purpose.

The proposed approach has its root in the design space exploration technique where at the outset a single cluster of all the keys are taken in a compact layout to evaluate the theoretical text entry rate. Further exploration is made for seeking rate comparison by segmenting into bi and tri cluster layouts respectively to identify which layout gives a marginal improvement. The keyboard structure obtained by this process forms an input to the Optimal Letter Arrangement algorithm which involves a pool of such keyboards with identical geometrical structure but uniqueness in letter placing. The algorithm runs till the pool gets emptied and from the set of removed keyboards the one which has the highest theoretical text entry rate is chosen.

II. Exploring Design Space

A. *Mono cluster approach*

In the mono cluster approach, all the vowels and the consonants are clubbed together excluding the numerals and the other keys. By allowing the users to use the layout we are able to measure the average text entry rate and then try and better the words per minute count. The practical text entry rate in this case for a novice user come up to 6.784 wpm(words per minute).

B. *Bi cluster approach*

In this approach, the vowels and the consonants are clubbed in two different groups excluding the numerals and the other keys. By allowing the users to use the layout we are able to measure the average text entry rate and try better the words per minute count as compared to the values obtained in the mono cluster approach. The practical text entry rate obtained in this case for a novice user come up to 8.144 words per minute.

C. *Tri cluster approach*

In this approach the vowels and the corresponding vowel-marks are grouped together whereas the consonant cluster is subdivided into two separate groups separated by the space key. By allowing the users to use the layout we are able to measure the average text entry rate and try and better the words per minute count as compared to the values obtained by applying the bi cluster approach. The practical text entry rate obtained in this case for a novice user come up to 10.4799 words per minute.

III. Fitness function measure

The observation based on the tri-cluster approach is based on the first hand use of the layout as designed. The arrangement of the letters occupies slightly greater space as compared to the mono cluster layout with highly improved results for the novice users. The space occupies however varies marginally with bi cluster approach but with a better text entry rate in the later case.

A. *Applying the Fitts' Digraph model for the Virtual Keyboards*

The model predicts user performance by summing the Fitts' law movement times (MT) between all digraphs weighted by the frequencies of occurrence of the digraphs. The use of Fitts' law made it possible to estimate performance in absolute terms. According to Fitts' hand movement law, the time taken to type a key say K_i to typing a key say K_j where i is not equal to j is given by:

$$MT_{ij} = a + b * ID \qquad (1)$$

Where a and b are the intercept and slope respectively and ID is the Fitts' index. We choose a = 0 and b = 1 / 4.9 sec/bit

$$ID = \log_2(A_{ij} / W_{ij} + 1) \qquad (2)$$

The average movement time is defined accordingly as:

$$MT_{avg} = \sum\sum MT_{ij} * P_{ij} \qquad (3)$$

Where P_{ij} is the probability of occurrence of the letters K_i and K_j respectively. If both the keys are the same then i will be equal to j, under such a scenario the following equation holds

$$MT_{i=j} = 1 \qquad (4)$$

In case of equation (4) Fitts' law is not applicable. Here there is the involvement of the same key so MT_{ij} is the time interval between the current press and the next press. The value is assumed to be one. Taking the reciprocal of the average movement time yields the average number of characters per second, which is transformed into words per minute.

$$CPS_{max} = 1 / MT_{avg} \qquad (5)$$

$$WPM_{max} = (CPS_{max} * 60) / 5 \qquad (6)$$

Assuming there are five characters in a word.

B. *Hick-Hyman's law*

As in Fitts' law the reciprocal of the Fitts' slope coefficient is denoted as b' while the intercept is denoted as a'. The slope co-efficient is called the bandwidth and is measured in bits per second. Bandwidth in this context is the rate at which humans process choices. The reciprocal of the slope in the Hick-Hyman law lies in the range 5 to 7 bps. Since the lower bound is to be searched for, we assume that the slowest choice processing speed is appropriate, and set b' = 0.2 seconds per bit. The response time RT according to this law is given as:

$$RT = a' + b' * \log_2 n \qquad (7)$$

Where n is the total number of keys in the layout.

C. *Calculation methodology*

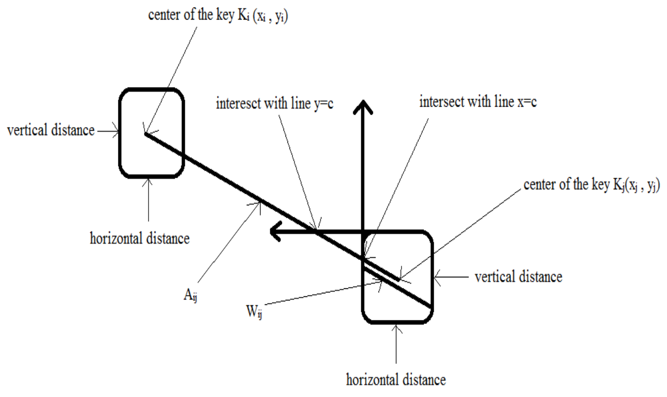

Figure 1. A_{ij} and W_{ij} calculation methodology

IV. Initial observation

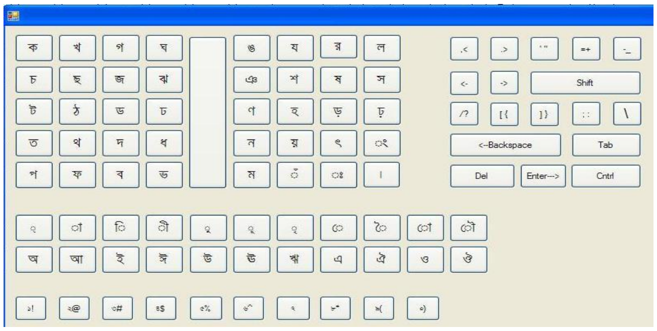

Figure 2. Tri cluster layout obtained

A. *Initial results*

The text entry rate count for an accustomed user turns out to be 25.18764 words per minute. This rate has been obtained by applying the Fitts' hand movement law. In case of novice users the Hick-Hyman's law is applied to calculate the response time. The minimum text entry rate comes out to be 7.1856 words per minute.

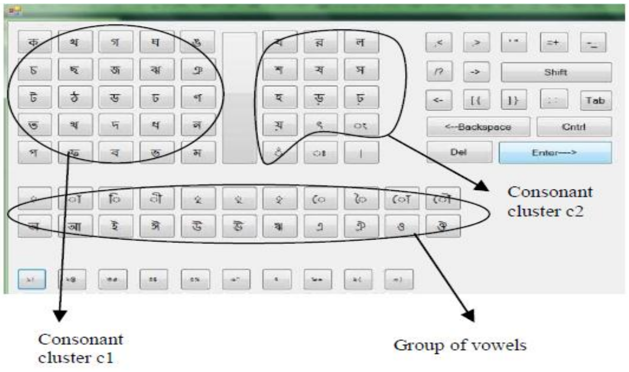

Figure 3. Modified tri cluster layout

The results obtained so far is by taking the modified tri cluster layout into consideration.

B. *Final modified layout*

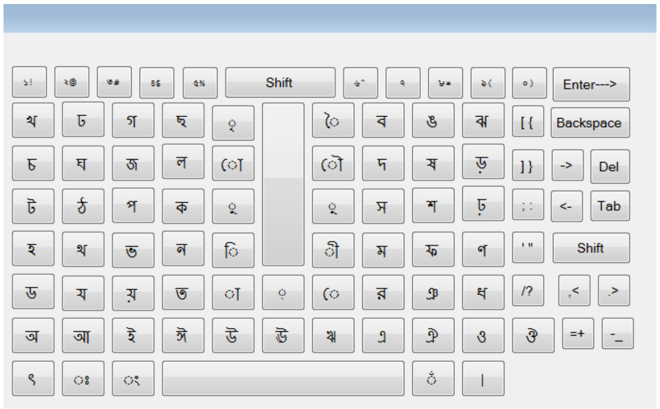

Figure 4. The final modified layout which forms an input to the Optimal Letter Arrangement algorithm

The modified layout is based on the fact that the space and the vowel-marks are placed centrally due to their frequent access among the conjuncts as obtained from the digraph probability results on a Bengali newspaper called Anandabazar Patrika. The consonants are divided into two separate clusters on either side of the space. The vowels occupy the final portion while the special symbols occupy the top segment.

C. *Threshold setting criteria*

A group of three hundred keyboards each having similar geometrical layout in accordance with Figure 4 but the co-ordinates of the letters are unique for each layout. After executing the theoretical text entry rates for each of those keyboards on a Anandabazar Patrika corpus it has been observed that out of three hundred, thirty keyboards posses a text entry rate of 31.046188 words per minute which means that a probability of 0.1 existing which is higher than any other rates produced.

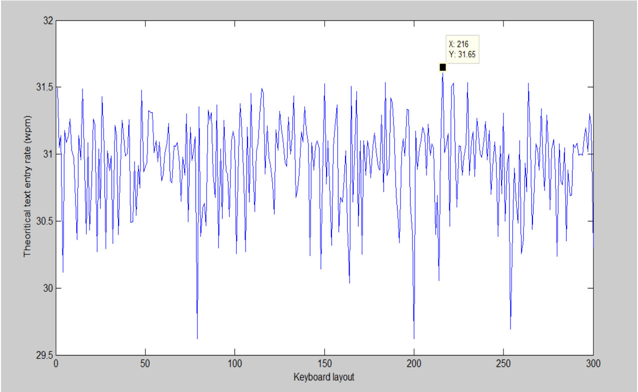

Figure 5. Ten percent keyboards having 31.046188wpm

The highest text entry rate obtained after testing is 31.65wpm amongst all the keyboards. It is thus inferred that if a keyboard in course of running of the Optimal Letter Arrangement algorithm crosses this value it is considered as fit to make a hop to the next generation. Otherwise a known layout which on 30 repeated trials manually by users crosses the limit of 31wpm or more 19 times. Thus, the probability of an unfit keyboard to attain fitness on inheriting this known layout is 0.633.

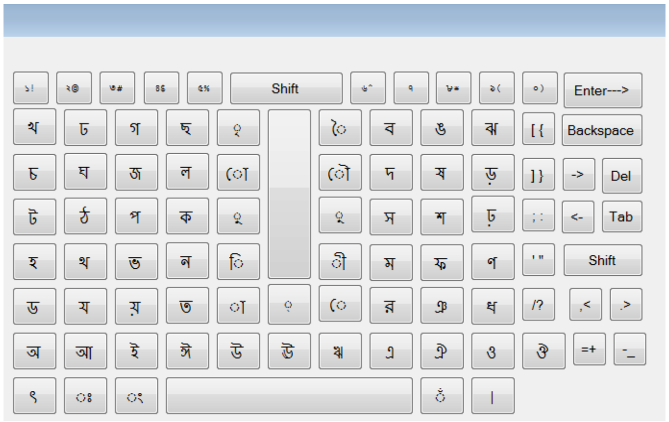

Figure 6. Predefined layout for unfit keyboards to acquire on being passed to the next stage.

V. Proposed approach

A. *Optimal Letter Arrangement algorithm*

Steps:

```
1. Initialize variables
        Set m = 1, sum = 0
        Set $t_j$ = 0, $round_j$ = 0

2. Write the consonants, vowels, vowel-marks, numerals
   in separate files each file consisting of similar type of
   characters.

3. The pattern of letter arrangement is different for each of
   the keyboards. This happens to be the start up current
   arrangement for each of the keyboards.

4. For each virtual keyboard $K_j$ j=1……k simultaneously
   do

   i) Initialization:
   a) $K_j$, loads its current_consonant_arrangement;
   b) $K_j$, loads its current_vowel_mark_arrangement;
   c) $K_j$, loads its current_vowel_arrangement;
   d) $K_j$, loads its current_number_arrangement

   ii) Randomization:
   a) Select two consonants from the consonant set
      randomly and swap the position of the letters between
      them in the layout. This swapping is done only between
      the consonants, both inter cluster as well as intra cluster
      since there are two consonant clusters;

   b )Select two vowel marks from the vowel mark set
      randomly and perform the interchanging of the
      positions. This interchanging is carried out only among
      the vowel marks in the layout;

   c) Select two vowels from the vowel set randomly and
      perform the interchanging of the positions. This
      interchanging is carried out only among the vowels in
      the layout;

   d) Select two numbers from the number set randomly and
      perform the interchanging of the positions. This
      interchanging is carried out only among the numbers in
      the layout;

   increment m;

   iii) Repeat Step 4(ii) until m = 5;

   iv) Initialize the array all_arr[100] ;

   v) Place all the characters which are randomly arranged
      that includes consonants, vowel marks, vowels and
      numbers into all_arr and fill the remaining cells of the
      array with the rest of the characters that comprises the
      remaining keyboard.

   vi) The Keyboard displays itself according to the changed
      consonant, vowel marks, vowels and number
      arrangement.

   vii) for p varying from 0 to 99 in all_arr[100]
          for q varying from 0 to 99 in all_arr[100]
          a) digraph = concatenation of all_arr[p] and
             all_arr[q];
          b) $A_{ij}$ = Euclidean distance between the source key
             all_arr[p] and target key all_arr[q];
          c) if (all_arr[p] == all_arr[q])
                 Wij = 0;
                 $M_{ij}$ = 1;
             else
                 $W_{ij}$ = the width of the target key all_arr[q] in
                 the axis of the motion;
                 a =0 , b = 4.9;
                 $M_{ij} = a + b * \log_2(A_{ij}/W_{ij} + 1)$;
             end if
          d) count = the frequency of the digraph pattern in
             the whole corpus taken;
          e) $P_{ij}$= count/length of the corpus;
          f) sum = sum + $P_{ij}*M_{ij}$ ;
        end for
      end for

    viii)
        a) $MT_{avg}$ = sum;
        b) CPS = 1/$MT_{avg}$ ;
        c) WPM = (CPS*60)/5;

5. After the end of the whole process, the text entry rates
   for each of the k keyboards in the pool is evaluated say
   $w_1,w_2,……..,w_k$ and thereby a fresh pool of keyboards
   is obtained.

6. For any keyboard $K_j$ j=1…..k
   if ($w_j$ >= 31.046188 wpm) then
   $K_j$ is allowed to pass to the next level of iteration
   carrying the same changed arrangement so that the
   next random change occurs on the current arrangement
   obtained.
        a) current_consonant_arrangement = changed
              consonant arrangement;
       b) current_vowel_mark_arrangement = changed
           vowel_mark arrangement;
       c) current_vowel_arrangement = changed vowel
          arrangement;
       d) current_number_arrangement = changed number
          arrangement;
   else
   $K_j$ is subjected to the following changes:
       a) current_consonant_arrangement = pre-
          determined consonant arrangement;
       b) current_vowel_mark_arrangement = pre-
          determined vowel_mark  arrangement;

7. for each $K_j$ where j varies from 1 to 21
     if ($w_j$ >= $t_j$)
          if($w_j$ – $t_j$ <=0.15)
               $round_j$++;
                 if($round_j$ == 10)
                     Do not pass the keyboard for further
                     changes, remove $K_j$ from the pool;
                 end if
          end if
    end for

8. Repeat Step 4 to Step 7 till no more keyboards are left
   in the pool.

9. Sort all the removed keyboards in descending order
   according to their final theoretical text  entry rate and
   select the highest value producing keyboard.

10. End.
```

VI. Results

Obtained layout after the whole process produces a text entry rate of 32.25217752 words per minute.

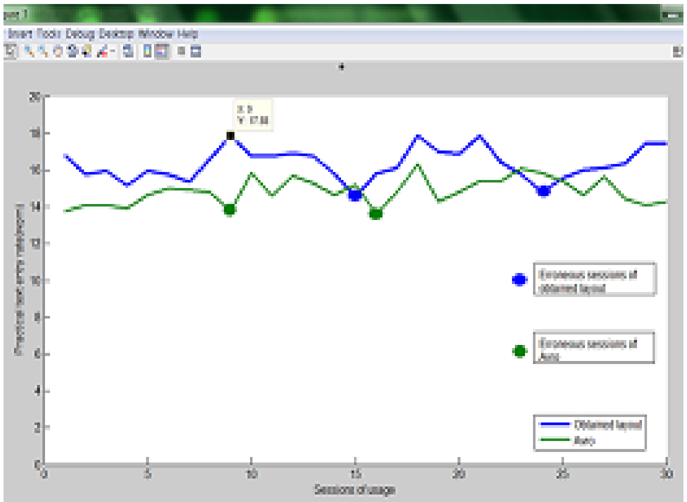

Figure 7. Comparison with the *AVRO* layout with the green curve representing the rates from *AVRO* while the blue curve is that of the finally obtained layout. The dots in the curves are the erroneous points.

Acknowledgment

The authors would like to express sincere thanks to fellow colleagues for their timely suggestion and advice in bringing out the desired work.

References

[1] R. William Soukoreff and I. Scott MacKenzie,” *Theoretical Upper and Lower Bounds Typing Speed Using a Stylus and Soft Keyboard* ”, Behaviour & Information Technology, 14,370-379.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Goldberg, D.E. Genetic Algorithms in Search. *Optimisation and Machine Learning*, 1989K. Elissa, “Title of paper if known,” unpublished.

[3] MacKenzie, I.S., and Zhang S.X. The design and evaluation of a high-performance soft keyboard. *CHI 99*, ACM Press (1999), 25–31

[4] Sayan Sarcar, Soumalya Ghosh, Pradipta Kumar Saha, Debasis Samanta IIT Kharagpur, “Virtual Keyboard Design: State of the Arts and Research Issues“, *Proceedings of 2010 IEEE Students' Technology Symposium 3-4 April 2010, IIT Kharagpur.*

[5] W. L. Dvorak A. Merrick N. L. Dealey and G. C. Ford, “Typewriting behavior.” *Wikipedia.*