2019 China-Qatar International Workshop on Artificial Intelligence and Applications to Intelligent Manufacturing (AIAIM)
January 1-4, 2019, Doha, Qatar

Adaptive Testing Tool for Students with Dyslexia

Ravisha Gupta
College of Engineering Pune, Shivajinagar, 411005 Pune, Maharashtra, India,
Email: guptaravisha17@gmail.com

***Abstract*—Dyslexia is a neurological and lifetime condition that is often inherited. One out of ten people have some or the other traits of dyslexia, and because of this around 20% face difficulties in reading and writing. Appropriate guidance provided to students with dyslexia at an early age can reduce the problems faced by them drastically. With massive strides in technology, a digital solution to this problem can prove to be more effective and efficient. For this purpose, a helping aid in the form of an android application is devised. It will help to fasten the learning process among these students who come across a wide variety of digital tools and are more prone to get attracted to such tools for learning than the standard pen-paper tools. This application will help the user to master writing by practicing alphabets, numbers and words.**

***Keywords*—Dyslexia; machine learning; Naïve Bayes algorithm; adaptive learning; scoring algorithm.**

I. Introduction

Dyslexia is a mental condition for which the need for diagnosis in human beings is not taken with utter seriousness as compared to other disorders. Even after the diagnosis, there is no defined treatment and many organizations are still exploring the optimum solution for it. Students find it difficult to read, write, spell, concentrate and organize things when they suffer with dyslexia [1]. The simple task to read a paragraph and comprehend it might seem very difficult to them, but when the same text is read out loud, then they are able to understand it completely. Hence, dyslexic learners find it difficult to perform simple tasks, but they are often talented in various other activities. Therefore, this ability is used for them to gain expertise in writing. Our Writing Tool Aid Application uses the concept of tracing and freehand writing on the device. The application has the following four modules in it:

1) The first module is *Learning Alphabets*. It provides the dyslexic students with an interface to learn alphabets and numbers by tracing them out by a stylus pen or their finger.
2) The second module *Learning Words*, has words categorized into three difficulty levels – Easy, Medium and Hard – for the user to trace and learn.
3) The third module *Freehand Writing*, provides a writing space to write the given word, alphabet or number with a stylus pen or finger.
4) The fourth module *Testing & Learning*, has various levels which tests the user in specific categories of words which are difficult for people with dyslexia to comprehend and analyze. This test follows adaptive style to generate the next level.

The scoring of the alphabets and words – traced by the user with the stylus pen or hand – in the first and the second module is done by using our *Efficient Scoring Algorithm*.

The adaptive testing used in the fourth module uses *Naïve Bayes Classifier Algorithm* [2] and *Machine Learning* technique to generate new levels according to the performance of the student in the previous level. Implementation of these two concepts as well as the Efficient Scoring Algorithm is the main focus area of this paper.

II. Problem Statement

With this paper, we seek to provide aid to dyslexic students who have trouble in writing. There are various types of words due to which writing becomes a strenuous task. Similar sounding words and visually confusing letters of the alphabet – like letters whose mirror image forms another valid letter – lead to erroneous spellings. For example, 'b' and 'd' can cause confusion to dyslexics. Elaborating on these lines, the problem can be broken down into the following four parts.

A. Familiarizing with Alphabets & Numbers

The root cause for difficulty in writing is due to poor understanding of alphabets – upper case and lower case – and numbers. The first step to tackle this problem would be to familiarize the user with these by involving them in the learning process. For this the user will trace the letters and numbers displayed as a bitmap with a stylus pen or finger.

B. Familiarizing with Words

People with dyslexia quite often get confused with the letters when they see them together in a word. Simple words like *bed* could be written as *deb*, though the user might know how to write *b* and *d* separately. This problem will be taken care of by making them trace words with different levels of difficulty.

C. Writing without Tracing

As the user proceeds with the steps of mastering how to write, the next task would be to write the given word without tracing. Here, the user will be required to write down the word displayed for a short time, along with an audio prompt. For this, they will use an external drawing keyboard as an input. For example *Google Handwriting Input* can be used to write the word using a finger or a stylus pen.

978-1-5386-6445-2/19/$31.00 ©2019 IEEE

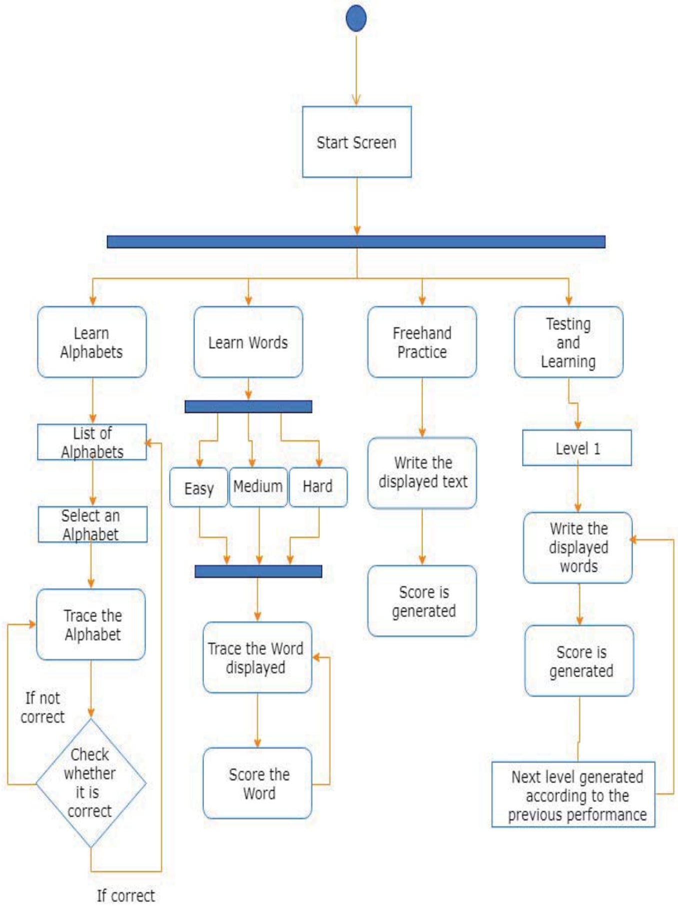

Fig. 1: Activity Diagram of the Writing Module

D. Adaptive Testing

It is difficult for the user to understand the mistakes which they commit while writing. So, the main aim of this section is to make the user understand its weak points and make them learn from their mistakes. As there are various categories [3] of words which dyslexic people find trouble in writing, our application will test the users on various levels by giving them words from different categories. The difficulty level of the words in various categories will depend on the mistakes made by the user in the previous level in those categories.

From this mobile application dyslexic users will learn how to write similar visual and auditory words. By taking the adaptive test, they will become aware of the type of words which they find difficult to write and will further help them to identify such categories. Hence, it will guide them to practice words from those categories.

III. Literature Survey

The teaching techniques that are used with students without dyslexia, generally do not work that well with dyslexic students. This is mainly because dyslexic students find it difficult to read, and in turn write, the alphabet. Therefore, to help them read and write better, it is necessary that techniques beyond the normal norms of teaching are used.

In the process of carrying out this research, different methods of teaching which can work best with dyslexic students were looked into. This paper, emphasizes on a possible combination of two such methodologies or models of teaching: Bloom's Taxonomy and The VARK Model.

A. Bloom's Taxonomy

The Bloom's Taxonomy [4] is a set of three models that were used to classify different educational learning techniques depending on their levels of complexity and specificity. Benjamin Bloom designed this model in 1956 along with collaborators Max Englehart, Edward Furst, Walter Hill, and David Krathwohl and published a framework for categorizing educational goals in the Taxonomy of Educational Objectives. The model designed by Bloom consisted six divisions namely:

1) Knowledge
2) Comprehension
3) Application
4) Analysis
5) Synthesis
6) Evaluation

Each division is like a block stacked on top of its predecessor. At the base, is knowledge, which here refers to recall of information. On top of knowledge, is comprehension, which means understanding what one recalls. Next up is application, because just understanding something won't be good enough if one can't apply it in the right situation. Going higher, comes analysis, which is the science of identifying and analyzing patterns. These patterns then need to be synthesized so as to improve what was applied in the first place, and that is why the synthesis block lies on top of analysis. At the top most position, according to Benjamin Bloom, lies the block of evaluation, which stands for assessment of the outcomes after passing through all the blocks below.

B. The VARK Model

The VARK Model [5] designed by Neil Flemming stands for Visual, Auditory/Aural, Reading/Writing and Kinesthetics (involving yourself in the learning process). The VARK Model basically highlights different teaching techniques that can be used by teachers to teach more effectively. Of these four types of techniques Visual techniques are implemented widely in almost all classrooms, however the other types of techniques are not implemented so often. Teachers always prefer to use learning modes that go hand in hand with student's behavior and their level of interaction.

C. Developing a Solution

Can these two models be intertwined to help the numerous dyslexic students in the world? That is exactly the question that this research is trying to answer. While there can be multiple ways to achieve the best of both the Bloom's taxonomy and the VARK model, this research focuses on one such way in particular. It aims at coming up with a solution that will involve the dyslexic user in way that will implement the six divisions defined by Bloom by using the VARK Model.

IV. Score The Accuracy of The Traced Handwriting

The first and second module of the application acquaints the user with upper-case and lower-case letters of the alphabet, numbers and as well as with words. This is done by displaying a bold bitmap covering the screen space. The user is then expected to learn by writing over it using a stylus pen or finger. By doing so repeatedly, the dyslexic user who often gets confused among the letters will gain better familiarity of these words and alphabets. This will enhance the capability to differentiate them in the first attempt.

The application prompts the user to keep trying until they have successfully traced out the letter or number or word within the displayed bitmap area. If the user writes outside the expected area, the attempt is wiped off and the application cues the user to write again without committing any more mistakes. Once the user traces it perfectly, the attempt is scored. The score is calculated in such a way that it accounts the number of mistakes committed, the number of attempts taken and the accuracy in tracing out the bitmap. For scoring the accuracy of the user, we have devised an *Efficient Scoring Algorithm* which uses the concept of pixel count and its comparison.

A. The Scoring Algorithm

For the purpose of demonstration, consider the bitmap of the word or letter to be in black color and the traced out handwritten attempt by the user to be in red color. Before the user traces out the displayed content, the algorithm will first get the pixel count of the letter or word displayed. To get the count of the black pixels, each pixel will be compared and checked with black color and the count will be appended. Hence, the area or pixel count of black is stored.

After the user has traced the letter or word or number, the algorithm will count in the number of red pixels in a similar manner. Now, the algorithm will check the percentage of the area covered by red color over black. 90% and above is given a full score and the scores decrease proportionally to the percentage of red pixels over black. A counter is also set for the number of times the user commits mistakes and is prompted to write again. Higher the counter value, the score awarded will be lesser.

$$Score \propto \frac{Percentage\ of\ the\ Red\ color\ over\ Black\ color}{No.\ of\ Reattempts} \tag{1}$$

B. The Efficient Scoring Algorithm

The above scoring algorithm is not efficient as it is not practical to compute each and every pixel on the screen. The number of pixels vary with the resolution of the screen. Higher the resolution, greater the number of the pixels are generated.

$$Pixels \propto R_{oS} \tag{2}$$

$$C_{Com} \propto R_{oS} \tag{3}$$

R_{oS} : Resolution of the Screen for a particular device

C_{Com} : Computational Complexities

This problem can lead to tedious computations and increase the complexities of the algorithm. To tackle this issue, we have devised an *efficient* way to count the pixels and compare the color of that particular pixel. For this, the drawing screen is divided into a grid using a *scaling factor*. This scaling factor plays a major role in determining the width and height of the grid cell.

Once the grid is formed, the *center pixel* of that individual grid cell is checked with the required pixel color. In this manner, the center of all vertical grid cells is checked for every horizontal cell. The black pixels of the displayed bitmap and the red pixels generated by tracing occupy a relatively different proportion of the screen. Because of this, their scaling factor also differs. To count the black pixels of the bitmap, the scaling factor is 50. To count the red pixels of the traced-out attempt, the scaling factor is 115. These constants were arrived at by calculations pertaining to the percentage of area of the screen occupied by the black bitmap and the red trace.

$$Grid\ Cell\ Size \propto \frac{1}{Scaling\ Factor} \tag{4}$$

As the scaling factor of black is smaller than that of red, the grid formed to calculate the number of black pixels will be larger than the grid formed to calculate the number of red pixels. This is required because the area covered by black pixels created by the bitmap on the screen is greater than the area that will be covered by red pixels by tracing, on the screen.

For the formation of the grid, the width and height of the individual cell is to be determined. x_{step} is the horizontal dimension of the cell andy_{step} is the vertical dimension of the cell in the grid.

$$x_{step} = \frac{Screen\ Width}{Scaling\ Factor} \tag{5}$$

$$y_{step} = \frac{Screen\ Height}{Scaling\ Factor} \tag{6}$$

Once the grid is formed, the center pixels of all the cells in the grid are checked for the required pixel color. For this, we first initialize coordinates of the first and last pixel using (5) and (6).

$$x_{init} = \frac{x_{step}}{2} \tag{7}$$

$$y_{init} = \frac{y_{step}}{2} \tag{8}$$

x_{init} : 1st Pixel's X-coordinate
y_{init} : 1st Pixel's Y-coordinate

$$x_{end} = \text{Screen Width} - \frac{x_{step}}{2} \quad (9)$$

$$y_{end} = \text{Screen Width} - \frac{y_{step}}{2} \quad (10)$$

x_{end} : Last Pixel's X-coordinate
y_{end} : Last Pixel's Y-coordinate

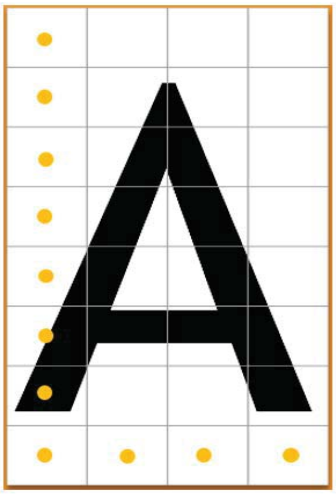

Fig. 2: Grid Formed by the Scoring Algorithm

After the initialization of the coordinates of the first grid cell pixel, the center pixel of all vertical cells in its column are checked. After completing the first column, it moves to the next one. This procedure continues till the last column of the grid is reached. The same task is followed for counting both black and red pixels but using different scaling factor to form the grids. In this way the pixel count for both the colors is done using the *Efficient Scoring Algorithm*.

Hence, the final score is calculated by:

$$P = \left[\frac{Number\ of\ Red\ Pixels}{Number\ of\ Black\ Pixels} \times 100 \right] \quad (11)$$

P : Percentage of Red pixels over Black pixels

$$\boxed{Final\ Score = P - [(Number\ of\ mistakes) \times 5]} \quad (12)$$

V. Adaptive Testing and Learning

By the time the user has completed the first three modules, they would have gained quite a lot of familiarity with the words and alphabets by tracing and writing them in the freehand mode. At this point they need to know the kind of mistakes they commit and practice more so that they can improve and learn from them.

For this purpose, we have developed an Adaptive Testing [6] module with the help of Naïve Bayes Classifier Algorithm and used the Machine Learning technique. There will be categories of words defined in which the dyslexic users face difficulty. Each category will have three sets of difficulty levels, which are – 'Easy', 'Medium'and 'Hard'. This module uses Naïve Bayes Algorithm to evaluate the difficulty factor of the next level according to the mistakes the user makes in the previous level and also to determine the expertise of the user in the given categories on the basis of the performance in various levels.

A. Flow of the Module

The test will display words to be written by the dyslexic user one by one, and the user is expected to write the words correctly in the first attempt. There will be five categories (A, B, C, D and E), and each category will have words of three difficulty levels. By default, the first level will display fifteen words of medium difficulty. The test will have the following defined categories:

1) Similar Auditory Words (Category A)
2) Words causing Visual Confusion (Category B)
3) Inversion of I and E (Category C)
4) Inversion of Words/ Anagrams (Category D)
5) Words with Addition/Omission of Letters (Category E)

B. Capturing User's Test Data

A table is created to capture the performance of the user in the various levels of the adaptive test. For every level the user attempts, a table is created. The tuples in the table consists of:

1) Question Number
2) Question
3) Category
4) Difficulty level
5) User Input
6) Result
7) Normalized Values
8) Next Level

The module prompts the user to write fifteen words, out of which five will be from pure categories and ten words will be from hybrid categories. A word from hybrid category could fall under, for example, both the categories A and B. Therefore, a level will have words from the categories A, B, C, D, E, AB, AC, AD, AE, BC, BD, BE, CD, CE and DE.

The Table I below illustrates an example of a user with expertise in the following categories ranked as shown:

A) Similar Auditory Words – Fair
B) Words causing Visual Confusion – Very Good
C) Inversion of I and E – Poor
D) Inversion of Words/ Anagrams – Very Poor
E) Meaningful Words with Addition/Omission of Letters – Good

After the user answers the first level, the next level is determined by using Bayesian Classification.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (13)$$

P(A) : Probability of occurrence of event A
P(B) : Probability of occurrence of event B
P(A|B) : Probability of A conditioned on B

TABLE I: SAMPLE TABLE FORMED AFTER THE USER TAKES THE ADAPTIVE TEST

Question No.	Question	Category	Difficulty Level	User Input	Result	Normalized value	Next Level
1	pat	A	Medium	qat	Incorrect	0.50	Medium
2	bed	B	Medium	bed	Correct	1.00	Hard
3	heir	C	Medium	heir	Correct	0.00	Easy
4	pit	D	Medium	tip	Incorrect	0.00	Easy
5	pain	E	Medium	pain	Correct	1.00	Hard
6	bat	AB	Medium	bat	Correct	0.75	Hard
7	zest	AC	Medium	sezt	Incorrect	0.25	Easy
8	sign	AD	Medium	sign	Correct	0.25	Easy
9	pizza	AE	Medium	pizza	Correct	0.75	Hard
10	queue	BC	Medium	qeueu	Incorrect	0.50	Medium
11	pram	BD	Medium	pram	Correct	0.50	Medium
12	trip	BE	Medium	trip	Correct	1.00	Hard
13	sassy	CD	Medium	sazzy	Incorrect	0.00	Easy
14	chief	CE	Medium	chief	Correct	0.50	Medium
15	drain	DE	Medium	rain	Incorrect	0.50	Medium

P(B|A) : Probability of B conditioned on A

TABLE II: THE BAYESIAN VALUES OF CATEGORIES

Category	Total No. of Right Answers in a Level (R)	Total No. of Wrong Answers (5 - R)	Bayesian Values
A	3	2	0.200
B	4	1	0.267
C	2	3	0.133
D	2	3	0.133
E	4	1	0.267

TABLE III: NORMALIZED AND BAYESIAN VALUES OF THE FIVE CATEGORIES

Category	Bayesian Value	Normalized Value
A	0.200	0.5
B	0.267	1.0
C	0.133	0.0
D	0.133	0.0
E	0.267	1.0

With reference to Table I, the total number of correct answers in this set are 9 and incorrect answers are 6. Now, the Bayesian Value of A for the correct answers after Level 1 is calculated in the following way:

$$P(A|Correct) = \frac{C(A) \times P(k)}{k} \tag{14}$$

C(A) : Number of Correct answers in A and its hybrid
P(k) : Probability of total number of correct answers
k : Total number of correct answers

$$P(A|Correct) = \frac{3 \times \frac{9}{15}}{9} = 0.20 \tag{15}$$

In the similar way, the Bayesian Values of all the categories are calculated. Table II shows the final value for each of those categories.

Now the module normalizes the Bayesian value for the non-hybrid categories by using the following formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{16}$$

X' : Normalized value of the Category
X : Bayesian Value of the Category
X_{max} : Maximum Bayesian Value
X_{min} : Minimum Bayesian Value

From Table II, $X_{max} = 0.267$ and $X_{min} = 0.133$. Hence, Normalized values for the Categories are:
A = (0.200−0.133) / (0.267−0.133) = 0.067 / 0.134 = 0.5
B = (0.267−0.133) / (0.267−0.133) = 0.134 / 0.134 = 1
C = (0.133−0.133) / (0.267−0.133) = 0
D = (0.133−0.133) / (0.267−0.133) = 0
E = (0.267−0.133) / (0.267−0.133) = 0.134 / 0.134 = 1

To calculate the normalized value for a hybrid category, the module takes the arithmetic mean of the normalized values of those two individual categories.

$$\frac{X'(Category\ 1) + X'(Category\ 2)}{2} \tag{17}$$

Therefore, on calculating, the normalized value of the hybrid categories is shown in Table IV.

TABLE IV: NORMALIZED VALUES OF THE HYBRID CATEGORIES

Category	Normalized Value
AB	0.75
AC	0.25
AD	0.25
AE	0.75
BC	0.5
BD	0.5
BE	1.0
CD	0.0
CE	0.5
DE	0.5

TABLE V: RANGES ACCORDING TO WHICH THE DIFFICULTY OF THE NEXT LEVEL IS DECIDED

Normalized Score Range	Next Level Difficulty for that Category
0.00 − 0.33	EASY
0.34 − 0.66	MEDIUM
0.67 − 1.00	HARD

The next level will be determined on the basis of the range defined below:

This module learns from the mistakes that the user commits. It recognizes the categories in which the user is weak and further helps them to learn by testing them from the same category. So, when the user performs poorly in a particular category, the next level will contain words of easier difficulty from the same category. It will keep testing them in those categories, until they improve and are ready to attempt the words with higher difficulty. In this manner, the successive levels of the adaptive test are generated. This will help the user to recognize their weak areas and will prompt them to strengthen and learn by practicing more.

VI. CONCLUSIONS

This paper highlights the issues faced by dyslexic users and how we can use their strengths to make the writing process easy. The *Writing Tool Aid* application includes four modules for this purpose. The *Learning Alphabets* module helps the user to strengthen the foundation by asking them to trace the letters and numbers. The *Learning Words* module also uses the tracing technique and helps the user to gain clarity in writing different letters together which can cause confusion. These two modules use the *Efficient Scoring Algorithm* to score the attempts of the user. This algorithm uses a different *scaling factor* for colors which is proportional to their coverage on the screen. This factor is used to divide the screen into a grid and the score is computed. The *Freehand Writing* module displays words along with an audio output, and the user is supposed to write on the screen with a stylus pen or finger and in the end a score is provided. The *Testing & Learning* module uses *Naïve Bayes Classifier Algorithm* and *Machine Learning* technique to create an adaptive test. This helps the user to understand their weak points and helps them in improving as they are prompted to clear the levels which provide words from the same category, but of easier difficulty level.

This application is a self-learning tool and it will motivate the people with dyslexia to involve themselves in improving in the skill of writing.

REFERENCES

[1] Understood for learning and & attention issues, Signs of Dyslexia at different ages, *https://www.understood.org/en/learning-attention-issues/signs-symptoms/could-your-child-have/checklist-signs-of-dyslexia-at-different-ages*, 2016, Online; accessed 5-March-2018.

[2] William M. Bolstad, *Introduction to Bayesian Statistics*.

[3] Dr. Vincent Goetry and Dyslexia International, Common patterns of mistakes and behaviours, *https://www.dyslexia-international.org/ONL/EN/Course/S2-3-2.htm*, 2017, Online; accessed 8-March-2018.

[4] Patricia Armstrong, Bloom Taxonomy, *https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/*, 2017, Online; accessed 2-April-2018.

[5] Jacqui Murray, The VARK Model of Teaching Strategies, *http://www.teachhub.com/vark-model-teaching-strategies*, 2015, Online; accessed 2-April-2018.

[6] S. Agarwal, N. Jain and S. Dholay, "Adaptive Testing and Performance Analysis Using Naive Bayes Classifier," *Procedia Computer Science*, vol. 45, pp. 70-75, 2015.