REAL LIFE PILOT SOLUTION WITH ARTIFICIAL INTELLIGENCE FOR DISABLED COMPUTER USERS

Karim Ouazzane, Markson Aigbodi
Faculty of Life Sciences and Computing, London Metropolitan University, T10-03 Tower Building,
London Metropolitan University, 166-220 Holloway Road, London, N7 8DB
{k.ouazzane, m.aigbodi}@londonmet.ac.uk

Jun Li
Department of Oncology, University of Oxford
Jun.li@oncology.ox.ac.uk

ABSTRACT

A user's typing stream contains all the information of user's interaction with computer by using QWERTY keyboard, which may include user's vocabulary, typing habit and typing performance. This paper is intended to develop a real life pilot solution for disabled people to use computer keyboard more effectively. Also, in this paper an extensive disabled user investigation on the performance of computer QWERTY keyboard is carried out. Then a Focused Time-Delay Neural Network model with extendable input and hidden layer neurons to analyze plain text and user's historical typing data is fulfilled. Finally, a development of a pilot application as an English input method is introduced. Approximately 50% FT Hitting Rate is obtained from experimental results which are considered as encouraging results and can be applied to symbol prediction and correction.

KEYWORDS

QWERTY keyboard, Focused Time-Delay Neural Network, English input method, Unary Coding, Hitting Rate

1. INTRODUCTION

Computer users with motor disabilities or cognitive problems may have difficulties in accurately manipulating the QWERTY keyboard. For example, for motor disability this may be seen in a form of tremor owing to a certain disease such as Parkinson's or any other factors (e.g., reduced range of hand motions due to Arthritis) [1], while cognitive problems usually are caused by loss of the ability to process, learn and remember information – for example, Dyslexia can cause significant problems in remembering even short sequences of numbers in the correct order.

These types of disability frequently cause typing mistakes, which haven't been well solved by current solutions [2] [3], not to mention problems people may have with several symptoms, which cannot be categorized as common types of symptoms. Although alternative input devices or software such as keyguard and Dasher [2] are available for use, none of them prove more efficient or comfortable than the current QWERTY keyboard. Some work associated with standard keyboard has been developed such as Windows' Accessibility Options and ProtoType [3], however the solution towards typing difficulties faced by disabled people hasn't been achieved.

N-gram prediction model [4] is a type of probabilistic model for predicting the next item in a sequence. It is widely adopted in natural language processing. But most current language modeling researches have been using samples collected from a large corpus. It has been argued that the corpus text is not a representative of user language, and it ignores the editing process and does not capture input modalities [5].

This research explores a Focused Time-Delay Neural Network (FTDNN) model [6] [7] to predict user's typing intention within a Virtual Key Code character set based on the historical typing data from Windows users. N-gram prediction can be achieved by using adjusted time delay neural network model and correction can be achieved in the same way by considering the correction as a type of predictions, which produces the right symbol based on the inaccurate historical data. Therefore, user's input context can be checked in sequence along with user's typing process. In the end, a pilot application as an English input method is introduced based on the model results.

2. PILOT MODELING USING NEURAL NETWORKS

2.1 User investigation

In this research, about 27 people have been interviewed. Both, old and disabled people were involved. Their major performance can be classified as four categories as illustrated below;

Motor disability

1. Long key press: this occurs when an alphanumeric key is unintentionally pressed for longer than the default key repeat delay.
2. Modifier keys: for example, "Shift"+ "a". One-hand typists in particular may find it difficult to press two keys at once.
3. Additional keys: some users often press keys adjacent to the intended keys.

Dyslexia

1. Miss letters or add letters: for instance, "student" -> "studnt".
2. Letters reverse: for instance, "student"->"studnet".
3. Similar word errors: for instance, "dose"->"does".

Unfamiliar with computer

1. Difficult to find keys: especially function and punctuation keys (e.g. F12).
2. 'Enter' key puzzle: some computers are with no "enter" or "shift" printed on the key surface, and therefore this makes it difficult for old people.
3. Compound keys problem: due to different definitions in distinct software, compound keys bring troubles to many users.

Others' performance

1. Miss words: leave out words in the typed sentences.
2. Mix words: reverse words in a sentence.
3. Mix lines: if there are some similarities between two or more lines (for example, same words), the user could mix lines.

2.2 N-gram prediction and FTDNN

For this research, a three layer FTDNN network shown in Figure 1 with twenty-seven input neurons, twenty-seven output neurons, extendible numbers of hidden layer neurons and extendible numbers of time delays is designed. This network is named as Focused Time-Delay Neural Network N-gram model, which combines Focused Time-Delay Neural Network and n-gram prediction method.

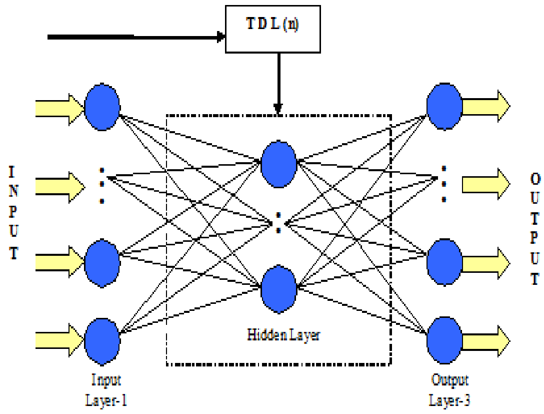

Figure 1 Architecture of Focused Time-Delay Neural Network N-gram Model

In order to evaluate the experimental results, two concepts are introduced here namely, Hitting Rate (HR) and First Three (FT) Hitting Rate, which refer to the probability falling into the top ranking and top three ranking respectively.Two datasets have been used in this research: dataset one – novel 'Far from the Madding Crowd (1874)' and dataset two – Disability Essex Helpline keystroke log. The novel was written by Thomas Hardy [8]. It has been used as a testing sample by some compression algorithm researchers. The version used here is from Calgary Corpus [9] with a size of 751kb. A piece of data (100k) is extracted from dataset one. It is subsequently divided into training data, validation data and testing data. The computer of Disability Essex Helpline has been used as a question recording, database querying and email tool by a disabled volunteer. As discovered in the log, the typing mistakes are predominately about adjacent key press and prolong key press errors. KeyCapture software [10] is modified to record user's typing log. It runs under Windows background and records keystrokes without interfering with user's work. A piece of software is designed to extract Virtual Key Codes [11] values according to the keystroke status from dataset two. In this research only the most commonly used fifty-three Virtual Key Codes are adopted, others such as arrows and functional keys are deleted from the log.

As raw data, the gathered datasets need to be preprocessed before it can be used by the FTDNN N-gram model to simulate the probability of each predicted symbol. Let's suppose to model a data sequence $C = \{s_1 ... s_i ... s_n\}$ on an alphabet basis of size $A = \{a, ..., z\}$, where $s_i \in A$. A sample of unary code is shown in Table 1,

Table 1. Unary code sample

Alphabet	Coding
A	1
B	01
C	001
D	0001
E	00001

Unary coding is an entropy encoding that represents a symbol by using *n−1* zeros followed by a one. ASCII coding requires less neurons (i.e. between 8 and 20 at input layer), but need to summarize all the possible output (>8bits).

A twenty-seven symbols set {*a…z, space*} is applied to dataset one based on the model shown in Figure 1. All capitals are converted to lowercase, while the other symbols are converted to space. Within the output a post-processing function is used to generate unary codes by ranking the twenty-seven outputs in a descending order. The maximum value of the output is converted into one and the rest values are converted into zeros for the calculation of First Hitting Rate.

During the training and testing of the FTDNN N-gram model related to dataset one, the numbers of grams – [*1, 2, 3, 5, 7, 9, 11, 13*] which are represented by time delays, and the numbers of hidden neurons – [*1, 2, 3, 5, 7, 9, 15, 25, 50, 100*] are cross-designed and implemented. Thereinto as the gram reaches *11* and the number of hidden neurons reaches *100*, the gram reaches *13* and the number of hidden neurons reaches *15* onwards, the memory of current system is beyond its limit. As a result, the experimental results are abandoned from *G-11 & H-100* onwards. As illustrated, the model uses a *27-n-27* three-layer structure. The experimental results related to dataset one plotted with First Hitting Rate and First Three Hitting Rate are shown below,

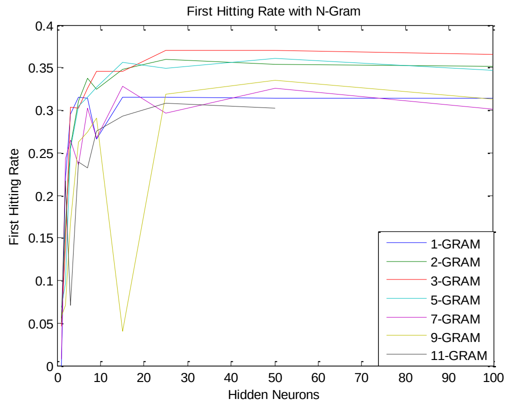

Figure 2. First Hitting Rate with N-gram

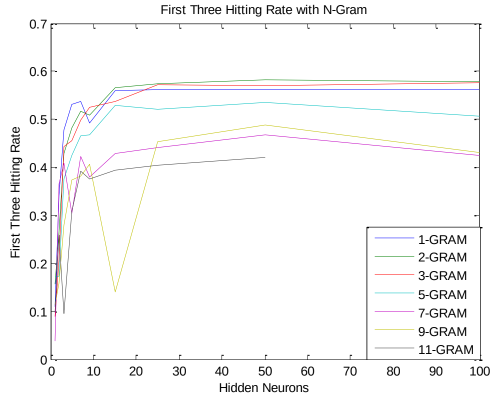

Figure 3. First Three Hitting Rate with N-gram

Fig 2 and Fig 3 show that *1, 2 & 3*-gram give the three best Hitting Rate (by winning in a small margin, *3*-gram gives the best First hitting rate and *2*-gram gives the best FT hitting rates), all of which show a better convergence toward the maximum Hitting Rate. Both figures illustrate the smaller Hitting Rates from *4*-gram onward. The results suggest that under the training sample, there would have been a best gram with certain number of hidden units to suit the prediction best. Subsequent to a certain increase, additional increase of gram or hidden unit doesn't help finding a good prediction. The figures also show that the number of neuron in hidden layer affects the model's learning ability and Hitting Rate. As suggested, the hitting rate in a hierarchy levels also can be used in ranking prediction.

Due to the limited learning ability of less number of hidden neurons as illustrated in the experimental results, the testing related to dataset two with one and two hidden neuron are ignored. And due to the memory limitation, the testing of *11*-gram and *13*-gram are abandoned. So, for typing stream dataset two, the chosen grams set is [*1, 2, 3, 5, 7, 9*] and the hidden neurons set is [*3, 5, 7, 9, 15, 25, 50, 100*]. The model uses a *53-n-53* three-layer structure by only changing the numbers of input and output neurons of figure 1. The experimental results related to dataset two plotted in First Hitting Rate and First Three Hitting Rate are displayed below,

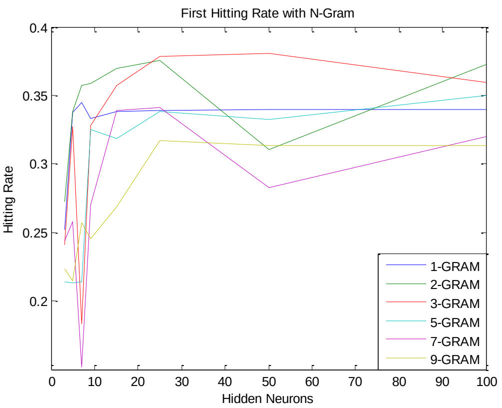

Figure 4. First Hitting Rate with N-gram

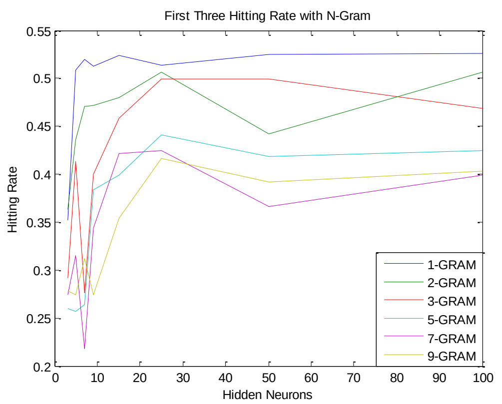

Figure 5. First Three Hitting Rate with N-gram

As shown in Fig 4 and Fig 5, *1*-gram has gained the maximum FT Hitting Rate – 53% and *3*-gram with fifty hidden neurons produces the maximum First Hitting Rate – 38.1%. Similar results have been obtained when test was carried out with dataset one: the lower grams (*1, 2&3*-gram) show a better solution with the FTDNN N-gram model prediction under current circumstance. Both datasets demonstrated a highly accurate prediction rate (FT Hitting Rate around 50%) with FTDNN model.

The experimental results can be used to predict users' typing intention. In practice a higher prediction rate could be obtained by combining the FT Hitting Rate with an English word dictionary. As the typing stream includes all the users' correction actions and the predicted next symbol could be 'delete' or 'backspace', the experimental results can also be used to improve users' current typing performance. Both tests (dataset one & two) show that a minimum number of hidden neurons are required in order to get a good hitting rate. The tests also show the gram uncertainty in terms of getting a best hitting rate; for example, in Fig 3, *2*-gram gives the best FT hitting rate while *1*-gram has the best FT hitting rate in Fig 5. Therefore, a combination of *1, 2* and *3*-gram is an optimal solution to keep a considerably high and stable hitting rate.

2.3 A self-learning Windows application

Visual C is used as a development tool based on Windows environment. IME (Input Method Editor) API is used to provide the system with a way to communicate with most editors. For efficiency reasons a unique database and its interface has been developed instead of using ODBC. An interface of pilot application based on the previous modeling results by using Notepad as an editor is shown in Figure 6.

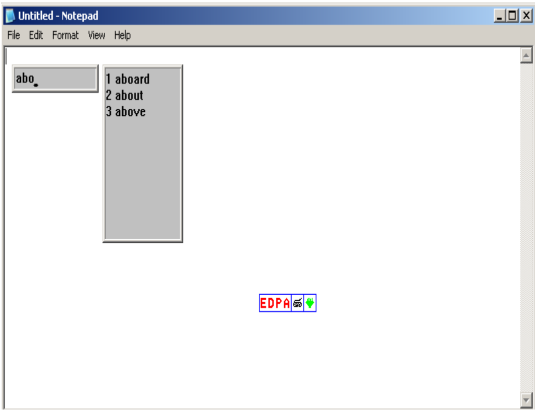

Figure 6. The user interface of the self-earning system using FTDNN N-gram model

The reason behind typing mistakes is complex. It depends on numerous factors such as, user mobility, computer environment and typing context, which require specific strategies to deal with. Further development of this application can focus on individual typing behaviour study and more featured algorithms integration.

3. CONCLUSION

This research provides a pilot solution for disabled people using computer keyboard by combining Focused Time-Delay Neural Network and n-gram modeling. It uses a Focused Time-Delay Neural Network model with extendible numbers of hidden layer neurons and extendible numbers of time delays to analyze plain text and user's historical typing data. The extendible numbers of time delays are simulated by n-grams. Approximately 50% FT Hitting Rate has been obtained from experimental results. In practice, the results can be applied to symbol prediction and correction. Thereafter, a pilot application based on Windows and IME API is suggested to demonstrate the modeling achievements.

Further research will include using a distributed representation method to preprocess the typing symbols, where each symbol will be represented by several features such as key distance, time stamp and symbol itself. In the future it is worthwhile to explore the application of FTDNN model to predict *l*-length string, so with the same *n*-gram input used in this research, more symbols can be predicted.

ACKNOWLEDGEMENT

The research was funded by Disability Essex [12] through Knowledge Transfer Partnership [13]. Thanks to Richard Boyd, Pete Collings and Stuart Kirk for helpful advice and discussions.

REFERENCES

1. Trewin S. and Pain H. (1998) 'A Model of Keyboard Configuration Requirements', *International {ACM} Conference on Assistive Technologies*, pp. 173-181
2. Inference Group of Cambridge, *Dasher*, Retrieved 25 January 2008 from http://www.inference.phy.cam.ac.uk/dasher/.
3. Sensory Software International Ltd, *ProtoType*, Retrieved 15 February 2008 from http://www.sensorysoftware.com/prototype.html.
4. *N-gram*, Wikipedia, Retrieved 26 November 2008 from http://en.wikipedia.org/wiki/N-gram.
5. Soukoreff, R.W. et al (2003), *Input-based language modelling in the design of high performance text input techniques*. Proceedings of Graphifcs Interface, pages 89-96, 2003.
6. Simon Haykin, *Neural Networks – A comprehensive Foundation 2nd ed*. Tom Robbins, 1999.
7. *Focused Time-Delay Neural Network (newfftd)*, The MathWorks, Retrieved 18 January 2009 from http://www.mathworks.com/ access/helpdesk/help/toolbox/nnet/dynamic3.html.
8. Thomas Hardy, *Wikipedia*, Retrieved 22 January 2009 from http://en.wikipedia.org/wiki/Thomas_Hardy.
9. Charles Bloom, *PPMZ–High Compression Markov Predictive Coder*, Retrieved 18 January 2009 from http://www.cbloom.com/src/ppmz.html and ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus/text.compression.corpus.tar.
10. William Soukoreff et al. *KeyCapture*, Retrieved 18 January 2009 from http://dynamicnetservices.com/~will/ academic/textinput /keycapture/.
11. *Virtual key codes* [online], available: http://api.farmanager.com/en /winapi/virtualkeycodes.html [accessed 05 February 2009]
12. *Disability Essex*, Retrieved 18 January 2009 from http://www.disabilityessex.org.
13. *Knowledge Transfer Partnership*, Retrieved 18 January 2009 from http://www.ktponline.org.uk/.

ISBN: 978-0-9853483-3-5 ©2013 SDIWC