A Fine-Grained Approach to Evaluating Recognition Errors when using Handwritten Text Input

Janet C Read, Chris Casey & Stuart MacFarlane

Dept. Computing, University of Central Lancashire, Preston, UK

{jcread, ccasey, sjmacfarlane }@uclan.ac.uk

Abstract:
It is usual to measure the accuracy of a recognition-based text entry system by using a correctness measure that is obtained by counting up the substitutions, insertions and deletions that have taken place between the original and the resultant texts. This paper presents evidence of situations where this sort of measure lacks clarity.
An alternative approach is described which scores errors that arise during handwriting recognition by considering how one character could be transformed into another. This results in a closeness matrix which is used to produce a Topological Distance Measure (TDM). This can be used to discriminate between results that appear to be the same when a traditional percentage correctness measure is calculated.
Applications for the matrix and the measure include usability evaluations, the design of user training and the identification of user errors.

Keywords: handwriting recognition, evaluation, errors, metrics, text input

1 Introduction

Human language technology covers a broad range of activities that share the goal of enabling people to communicate with machines using natural communication skills. This technology includes speech recognition, handwriting recognition and gesture recognition (Cole et al., 1997).

For the interface designer these recognition-based computer systems present unique challenges. The human machine interaction is changed by the use of a natural communication device and, in addition, the algorithms that are used for recognition often introduce errors that the user has to be able to discover and repair (Noyes, 2001).

The goal for the interface designer is to produce a human language interface that is error free, efficient, and easy to use. Appropriate metrics are needed to assess progress towards these goals and to assist in the diagnosis and repair of erroneous behaviour within the human – computer interface.

This paper begins with an overview of evaluation metrics for recognition-based interfaces and then describes the standard percentage correctness measure that is widely used for text input. Two alternative measures that have been used with speech recognition interfaces, phonological closeness and human reference standards are also examined.

Section four applies a Character Error Rate measure to handwriting recognition and uses an example to demonstrate a scenario where a more discriminating metric would be expected to produce different results.

Section five describes how a new metric for handwriting recognition based on the topological closeness of characters in the English alphabet was developed and evaluated by the authors. It does not apply this work to other alphabets; although this would be possible.

The usefulness of this new metric is then discussed and areas for future development and research are outlined.

2 Using Recognition - based technology for Text Input

Recognition-based technology can be used for command and control environments as well as for text entry. The evaluation of the success of a command control environment is significantly different from the evaluation of the success of a text input interface. This paper is only concerned with text input.

Text entry to a computer can be effected in many different ways. It may involve a keyboard or a mouse; it may involve a pen or a microphone. It is possible to classify text entry modes not only according to the hardware used, but also by considering the initial product of the text entry process. This may be ASCII text or human readable signals. In the latter case, there needs to be further processing by recognition software to produce ASCII text. The text entered at the computer may be copied from prepared text or may be composed at the interface.

Of the three recognition technologies (speech, gesture and handwriting), speech and handwriting have been extensively promoted for unconstrained text input (Daly-Jones et al., 1997), (Lorette, 1998).

Work on the evaluation of text input methods focuses on the three usability measures of efficiency (speed of input), effectiveness (error rate) and user satisfaction (Mackenzie and Soukoreff, 2002b). Efficiency is typically measured in characters per second, user satisfaction is measured by the use of questionnaires and by observation, and effectiveness is measured by making comparisons between two strings, the first being the presented text (PT) and the second the transcribed text (TT) (Mackenzie and Soukoreff, 2002a). For recognition-based interfaces, the term 'transcribed text' is a poor description of the final text string. It is the text that is generated from the recognition process that is compared to the presented text. However, in order to simplify the discussion within this paper, we have chosen to use TT throughout to represent the final text string.

It has been established that for recognition – based interfaces, user satisfaction has been shown to correlate with effectiveness; (Frankish et al., 1995) that is, the better the recognition rate of the technology, the more the user is satisfied. Thus, improved accuracy within a recognition – based interface will result in an improved experience for the user.

3 Measures of Correctness

The accuracy of the recognition process is typically measured by apportioning a percentage score to text after it has been through the recognition process. The standard metric for this is a percentage correctness measure which, intriguingly, is normally calculated as a percentage error rate but which is generally quoted as a correctness score!

3.1 Percentage Correctness Measures

The de-facto standard measures for accuracy are generated from two text strings; the presented text (PT) and the Transcribed text (TT). These two strings are compared and each 'error' in the generated text is classified as either an insertion (I), a deletion (D) or a substitution (S). This gives a numeric score that is divided by the number or words (or characters) in the prescribed text to give an error rate (E).

This measure can exist in two forms, as a word error rate (as is typically used in speech recognition)

> WER = (S + I + D) / N where N is the total number of words in the test set, and S, I, and D, are the substitutions, insertions and deletions.

or as a character error rate (as is typically used in handwriting recognition as well as in discrete text input such as that which is done at a keyboard)

> CER = (S + I + D) / N where N is the total number of characters in the test set, and S, I, and D, are the substitutions, insertions and deletions.

When classifying the errors, penalties or weightings are applied to the different types of error to ensure that a single substitution is preferred to the combination of a deletion plus an insertion. The American National Institute of Standards and Technology (NIST) uses a weight of four for substitutions and three for deletions and insertions choosing the least weighted score at each error. This is by no means standard.

In its original form, a character error rate can result in unexpected (and probably unrealistic) error rates. Work by (Mackenzie and Soukoreff, 2002a) has addressed some of these problems by applying a minimum string distance algorithm to matching words in the two text strings. Given *quickly* becoming *qucehkly,* application of the MSD algorithm results in an error rate of 3/8 = 37.5%. A standard PCM approach would have resulted in an error rate of 6/8 = 75%, given that all the last six letters are in the wrong place.

There are other problems relating to the length or size metric, N. It is possible to obtain an upper error rate in excess of 100% by basing the error rate on the number of characters in the presented text. (Soukoreff and Mackenzie, 2001) have suggested a modification wherein the greater value of N for the presented text and the transcribed text is used.

Further work by the same authors has produced an algorithm that can generate a CER automatically. This is one of the attractions of the CER measure; the relative ease with which it can be calculated (Mackenzie and Soukoreff, 2002a).

3.2 Phonological Scoring

The appropriateness of the PCM (WER) for speech has been challenged by a method known as phonological scoring as developed by (Fisher et al., 1995). This can be demonstrated in the following example.

Presented text;
'He called for a new start'
Generated text:
'He called foreign news the art'.

he	called	for	a	new	start
he	called	foreign	news	the	art

In PCM scoring, this would be seen as 4 substitutions i.e. an error rate of 66%;

However by attempting to minimize the differences between what was said and what was recognized it could be argued that the recognizer substituted *'foreign'* for *'for'*, missed *'a'*, substituted *'news'* for *'new'*, added *'the'* and then substituted *'art'* for *'start'*

he	called	for	a	new		start
he	called	foreign		news	the	art

Phonological closeness is about the 'sounds' of the phonemes (bits of speech), and about identifying those that are close. Thus, in the example above; '*for a'* is judged to be close to *'foreign'*.

3.3 Human Reference Standard

The human reference standard as proposed by (Cox et al., 1998) is a measurement based on human performance, whereby the question 'How much better is it than me?' would be asked, instead of 'How accurate is it?' The underlying issue is to see if the recognizer would be better at recognition than a human, and if so, by how much.

In their study (Cox et al., 1998) considered the recognition of isolated words using speech recognition. Using isolated words allowed them to remove the contextual help found in most speech recognition software that is known to distort research findings.

Listeners were given a 5100-word vocabulary and were required to select the word they had heard or reply 'unknown'. The spoken text was subsequently distorted by the addition of white and speech noise, and distortions of signals. The error rate by the listeners was shown to increase as the speech was degraded. The intention was to be able to claim that the recognizer was as good as humans on speech that had been degraded by a numeric factor.

The application of a human language model to handwriting deserves further research. It would be necessary to investigate methods for eroding the handwriting (e.g. adding random noise, distorting the characters or changing the resolution).

4 Application to Handwriting Recognition

In online handwriting recognition, the presented text can be captured as an ink file (human-readable). This is then 'recognized' by the recognition software and a machine-readable text file is produced. If handwriting recognition takes place off line, a bitmap or vector image (human-readable) of the intended text is converted into a machine-readable text file. The discussion that follows could apply to both modes of recognition.

Handwriting may be presented as characters or as words. It is usual for writers to use a mixture of cursive and discrete writing (Tappert et al., 1990) and so most recognition algorithms are initially based on character recognition.

Partly for these reasons, for a handwriting interface it is common to apply a Character Error Rate (CER), rather than a Word Error Rate when calculating recognition accuracy (Read et al., 2001), (Mankoff and Abowd, 1999). An example of the CER being applied to handwriting is given here.

In this example, the user was required to write the words;

beside the ocean there she sits-

This was then recognized by two different recognisers with the following results:-

Recognition A

(TT) renitle the ixean thene yhe sits-

CER = (7 + 1 + 0) / 32 = 0.25

(there are 7 substitutions and one insertion)

Recognition B

(TT) bosiide the occar tneveshe slts-

CER = (6 + 1 + 1) / 32 = 0.25

(there are six substitutions, one insertion and one deletion)

5 Topological Closeness

It was hypothesised that pairs of characters could be evaluated for closeness. This sort of evaluation would render the characters ‘a’ and ‘d’ to be closer to one another than the characters ‘f’ and ‘m’.

Using this information, it would be possible to carry out a more fine-grained evaluation of recognition accuracy, whereby for two text strings there would be two scores, one representing the ‘expected’ or ‘close’ errors and one representing the ‘unexpected’ or distant’ errors. This could then be used to produce a correctness measure that would differentiate between these two sorts of errors.

5.1 Development of a Closeness Matrix

When a recognition error occurs, it is most often caused by a relatively poor construction of the character by the user. There are other causes, including software and hardware failure, but these are in the minority (Read et al., 2002). The apparent similarity of some of the characters in the Latin alphabet does little to assist the recognizer in its differentiation.

We began by identifying how letter shapes could be made into one another by simple transformations. Seven transformations were considered. These were ‘grow’, ‘shrink’, ‘cut’, ‘join’, ‘bend’, ‘rotate’ and ‘mirror’.

Examples of these seven transformations are shown in Figure 1.

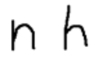
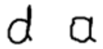
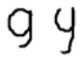
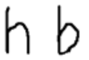
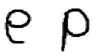
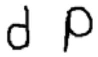
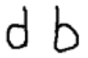
Transform	Example
Grow (g)	
Shrink (s)	
Cut (c)	
Join (j)	
Bend (b)	
Rotate (r)	
Mirror (m)	

Figure 1: Original Seven Transformations

A matrix of lower case letter representations was constructed and for each letter pair, the single or combined transformations that could make that change were noted. Figure 2 shows an extract from this matrix. In this, the transformation pair ‘mg’ represents a mirror AND a grow.

	a	b	c	d	e
a		mg	c	g	rg
b	ms			m	
c	j			jg	j
d	s	m	cs		rb
e	rs		c	rb	

Figure 2: Transformations for a to e

During the construction of the matrix, it became clear that there were many different ways of changing letter shapes to one another. For example, to change a ‘f’ into a ‘b’ one could join, shrink, rotate and mirror or one could mirror, join and shrink without the rotation. For each character pair, we chose the least number of transformations.

The generated matrix was compared with experimental data from our own work and from others. We considered the character pairs that had single transformations and compared these with reported mis-recognition pairs. These can be found in work by many authors including (Frankish et al., 1995), (MacKenzie and Chang, 1999) and (Tappert et al., 1990). The transformations mirror, bend and rotate failed to correlate whereas the single transformations grow, shrink, cut, and join accounted for 94% of the pairs of misrecognition.

To this end we have identified four ‘core transformations’ that we believe can be used to generate a table of ‘close characters’ in much the same way that phonological closeness uses ‘close sounds’. These transformations are grow, shrink, cut and join. Bend and rotate are hereafter referred to as weak transformations.

Using these four transformations, the character matrix was reconstructed and a score that represented the ‘minimum number of simple transformations’ needed to effect the change was derived for each character pair. With the four core transformations, the matrix for a to e is presented below.

	a	b	c	d	e
a	0	>1	1	1	>1
b	>1	0	>1	>1	>1
c	1	>1	0	>1	1

d	1	>1	>1	0	>1
e	>1	>1	1	>1	0

Figure 3: Closeness matrix for a to e

5.2 A Topological Distance Measure

We describe a new metric based on closeness that applies a lesser weight to close substitution errors than to distant substitution errors.

This measure relies upon the existence of a closeness matrix of the form found in Figure 3 above.

For two strings (PT) and (TT), the Topological Distance Measure (TDM) is described as

TDM = (CS/2 + DS + I + D) / N

Where CS is the total number of close substitutions as defined by a number 1 in the closeness matrix, DS is the number of distant substitutions as defined by a number greater than one in the closeness matrix, I is the number of insertions, D is the number of deletions and N is the number of characters in the presented text. 2 is an arbitrary constant that is used as a weight.

This is applied in the following way. The two text strings (PT) and (TT) are compared, and an MSD approach is used to line up the characters in such a way that the CER would be minimized. For each character pair a score of 0 (no substitution), 1 (close substitution) or >1 (distant substitution) is noted. For each single character, either an insertion or a deletion is recorded. An example of this procedure is shown below;

b	e	s	i		d	e		t	h	e
r	e	n	i	t	l	e		t	h	e
>1	0	**>1**	0	**ins**	**>1**	0		0	0	0

The following example shows how the topological distant measure is able to separate the two recognition results from Section 4.

(PT) beside the ocean there she sits-

Recognition A

(TT) renitle the ixean thene yhe sits-

TDM = (1/2 + 6 + 1 + 0) / 32 = 0.234

Here, there is one close substitution – this is the character pair 'r' , 'n'. The other substitutions are distant and there is one insertion.

Recognition B

(TT) bosiide the occar tneveshe slts-

TDM = (5 / 2 + 1 + 1 + 1) / 32 = 0.172

In this representation, there are 5 close substitutions, 'e', 'c' - 'n', 'r' – 'h', 'n' – 'r', 'v' and 'i', 'l'. There is one distant substitution, one insertion and one deletion (of a space).

6 Discussion and Further Work

It can be seen in the example in Section 5 that application of this metric can discriminate between two recognizers that have exactly the same performance with a CER metric. In addition, given a single closeness matrix, the TDM metric will give consistent, repeatable results.

6.1 Production of the Original Matrix

The closeness matrix in this study was only constructed for lower case Latin alphabet letters. It would be desirable to double the character set to include upper case letters as there are at least another 15 close pairs that could be identified from this extension. The inclusion of digits would expect to result in around 10 more close pairs. Punctuation symbols could also be included. An expand transformation and a shrink transformation would need to be introduced to differentiate between those characters that have identical shapes but represent different things. Examples of these include '*C*' and 'c' and 'S' and 's'.

6.2 Equality of the transformations

One area for further work is to determine whether or not the four transformations should be treated equally. The purpose of the metric is to reduce the penalty for errors involving 'close' characters. The probability with which people mis-interpret one character as another could be used to determine a weighting for the relevant transformation. This would replace the arbitrary weighting of 0.5 which was used in the metric.

6.3 The Usefulness of the metric and the matrix

We believe that this metric can assist in the production of useable handwriting recognition based applications. In the first instance, it can be used to more reliably measure how close a recogniser was to getting a character representation right. This information can be used to provide targeted training for the user who may just need to grow a portion of their writing to ensure better recognition results.

Secondly, the metric can be used to measure how an interface improves with time. Small improvements are more likely to result in improved TDM scores than in improved CER scores. A further application of the metric is for the diagnosis of user errors. Given an automated system for the calculation of error rates, the user is presented with a score that is simply the total number of errors. With the TDM, the errors could be divided into two types, this would allow the researcher to easily see the unexpected errors and he may use this information to check whether or not the user had written the presented text properly in the first place.

If the matrix could be used to decide whether the error was made by the user or the system (i.e. by assuming that close errors are made by the recogniser and unexpected or distant errors by the user, it could be used to inform the user of a problem with their letter formation. This would be a useful attribute of a system that was used by children.

6.4 Characters that fall apart

The closeness matrix can be extended to incorporate those instances when one character becomes two and when two characters become one. 'a' frequently becomes 'ci' and 'cl' frequently becomes 'd'. There are other regularly occurring situations like this. On first glance, it can be assumed that when 'a' became 'ci'what happened was that 'a' became 'c' and 'i' was inserted. This is not an accurate interpretation, it is extremely rare for users to insert extra characters when doing handwriting. These pseudo-insertions are almost exclusively caused by the splitting of a single character into two by the recogniser. In a similar way, it is unusual for users to fail to write characters; however, consecutive characters being recognised as one letter invariably present themselves as deletions.

This merge and split behaviour is relatively common in handwriting applications. It is a feature of those recognition engines that allow the user to use a combination of cursive and discrete text. Given that this is how most people write, this is a sensible model for recognition, but in evaluations, these doubling up recognitions will prejudice any error rate score as each merge / split error will be penalised twice. Applying a closeness measure to these would reduce the penalty to 0.5. As this would almost eliminate all the insertions and deletions, the researcher would be better able to see when the user had really inserted or deleted a character.

It may be possible to define an error measure for handwriting recognition entirely based on substitutions, some of which may replace a single character in the PT with multiple characters in the TT and *vice versa*

7 Conclusion

This paper has shown how traditional PCM measures are unable to separate dissimilar results. In particular they do not assist the researcher in assessing which is the better of two recognisers A and B where the substitution errors of A are mainly involving close characters and those of B are mainly involving distant characters. We have identified four core transformations, two weak transformations and have discussed the possibilities for some other transformations. The core transformations have been applied to the lower case characters of the Latin alphabet set and this has resulted in a closeness matrix.

This work has resulted in a metric for handwriting recognition that is more discriminating than CER and is therefore better able to support the diagnosis of errors and to highlight those areas that need improvement.

Future work has been described that will enlarge the character set for the closeness matrix. And will extend the set of transformations to include double letter transformations.

References

Cole, R., Mariani, J., Uskoreit, H., Zaenen, A. and Zue, V. (Eds.) (1997) *Survey of the state of the art in human language technology,* The press syndicate of the University of Cambridge, Cambridge.

Cox, S., Linford, P., Hill, W. and Johnston, R. (1998) Towards speech recognizer assessment using a human reference standard, *Computer Speech and Language,* **12**375 - 391.

Daly-Jones, O., Monk, A., Frohlich, D., Geelhoed, E. and Loughran, S. (1997) Multimodal messages: the pen and voice opportunity, *Interacting with Computers,* **9**1 - 25.

Fisher, W. M., Fiscus, J. G. and Martin, A. (1995),Further studies in phonological scoring In *ARPA Spoken Language*

*Workshop*Morgan Kaufmann, Austin, Texas, pp. 181 - 186.

Frankish, C., Hull, R. and Morgan, P. (1995),Recognition Accuracy and User Acceptance of Pen Interfaces In *ACM CHI'95*, pp. 503 - 510.

Lorette, G. (1998),Handwriting recognition or reading? Situation at the dawn of the 3rd Millenium In *International Workshop on Frontiers in Handwriting Recognition*Taejon, pp. 1 - 13.

Mackenzie, I., Scott and Soukoreff, R. W. (2002a),A Character-Level Error Analysis for Evaluating Text Entry Methods In *NordiChi2002*ACM, Aarhus, Denmark, pp. 241 - 244.

Mackenzie, I., Scott and Soukoreff, R. W. (2002b) Text Entry for Mobile Computing: Models and Methods, Theory and Practice, *Human-Computer Interaction,* **17**147 - 198.

MacKenzie, I. S. and Chang, L. (1999) A performance comparison of two handwriting recognizers, *Interacting with Computers,* **11,(3)** 283 - 297.

Mankoff, J. and Abowd, G. (1999),Error Correction Techniques for Handwriting, Speech, and other ambiguous or error prone systems In *Interact 99*.

Noyes, J. (2001) Talking and writing-how natural in human-machine interaction?, *International Journal of Human-Computer Studies,* **55**503-519.

Read, J. C., MacFarlane, S. J. and Casey, C. (2001),Measuring the Usability of Text Input Methods for Children In *HCI2001*, Vol. 1 Springer Verlag, Lille, France, pp. 559 - 572.

Read, J. C., MacFarlane, S. J. and Casey, C. (2002),Oops! Silly me! Errors in a Handwriting Recognition-based Text entry Interface for Children In *NordiChi 2002*Aarhus, Denmark.

Soukoreff, R. W. and Mackenzie, I., Scott (2001),Measuring Errors in text entry tasks: An application of the Levenshtein string distance statistic In *CHI 2001*, Vol. Extended abstracts of CHI 2001 ACM Press, New York, pp. 319 - 320.

Tappert, C. C., Suen, C. Y. and Wakahara, T. (1990) The State of the Art in On-Line Handwriting Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **12,(8)** 787 - 808.