Chapter 1

Implementation and Evaluation

1.1 E-Learning Module Evalutation

Generally, there are two main directions in the evaluation methodology for e-learning applications: the educationalist's approach and the software developer's approach. Therefore, the evaluation of an e-learning system is a complex task and requires optimisation work on the account of both the conceptual designer of an e-learning application as well as the software developer.

The e-learning part of the prototype is a sample module that is used to exemplify one usage scenario of the HWR engine. It mainly shows plausibility of the approach. A detailed analysis of the e-learning application using ISO9126 (**?**) is not useful, since the e-learning part of the software has not been optimised in any way. The focus of the thesis is not to implement an e-learning application, but to create an analytical handwriting recognition engine. It might be a prospect for future work[1] to optimise the e-learning part and build a fully-developed e-learning application for Japanese characters, but that would be outside the scope of this thesis. For these reasons, there will not be an evaluation of the e-learning module.

1.2 Evaluation of the HWR Engine

1.2.1 General Considerations for Evaluation

The performance of a recognition system can be measured in terms of speed, accuracy and memory requirement. While statistical systems offer high speed but have large requirements on memory, structural methods have lower speed but require only a smaller memory (**?**). The system developed and evaluated in this thesis is a structural system. It can be expected that the system has a relatively low performance speed, but moderate memory requirements. Additionally, since the system is not just a structural but an analytical recognition system, the speed might be even lower.

The factors memory requirements and speed are not of great interest in the context of this system. The system is an online system, but it exclusively performs single character recognition. The focus lies a lot more on a detailed analysis of one character, rather than the high-speed recognition of a stream of characters. The system is an interactive system. The recognition of a single character is an in-depth analysis of the structure of that character and returns a profound feedback to the user. Especially in an interactive learning context, the user is supposed to work with the system's feedback. Recognition speed is interesting for evaluation if the user enters a stream of characters. In that case recognition speed can be expressed as a factor that relates recognition speed with input speed. **?** (**?**) state that on-line recognition systems need only be fast enough to keep up with the writing. Further, they report average writing rates of 1.5-2.5 characters/s for English alphanumerics and 0.2-2.5 characters/s for Chinese characters. In a system that performs single character recognition the user has the impression of instant recognition.

Memory requirements are negligible in this system for a similar reason. The recognition of one character does not require much memory compared to the recognition of a stream of characters. Additionally, the main system engine does not run on a small mobile computing device with low memory capacity, but runs as a service on a standard PC. With the advanced memory capacities of today memory is not an issue. Nevertheless, even mobile devices are equipped with enough memory to enable the system to perform analytical character recognition.

For the reasons given above, the evaluation of the analytical handwriting recognition engine will be limited to different types of accuracy measurements.

[1] See section **??**

1.2.2 Evaluation of Other Systems

The recognition rates reported in the literature are shown in figure 1.1 borrowed from **?** (**?**). As a general trend it can be noted that the recognition rate of most systems lies between 85% and 95%. They believe that it is possible to achieve a recognition rate of up to 98% for regular scripts. On fluent scripts, however, they regard it as difficult to achive a recognition rate above 90%. The systems marked with an asterisk in figure 1.1 perform recognition for Chinese or Japanese characters. All of their recognition rates lie below 90%. That performance measure sets a general context in which the prototype system developed in this work might be arranged.

Source	Method	#category	Style	#learning	#test	Rec. rate
Liu'91 [75]	struct	6,763	flu-regular	N/A	N/A	90%
Kawamura'92 [45]	statis	2,965	careful/free	380 PC	20 PC	94.51/91.78%
Lin'93 [65]	struct	5,400	regular	1 PC	10 PC	87.4%
Liu'93 [76]	struct	13,000	flu-regular	N/A	N/A	93%
Hamanaka'93 [28]	statis	1,064	regular	54,028	52,944	95.1%
Chou'94 [16]	struct	5,401	regular	3 PC	17 PC	94.88%
Wakahara'95 [133]	struct	2,980	careful/free	120 PC	36 PC	97.6/94.1%
Lay'96 [60]	struct	5,401	regular	N/A	5 PC	96.35%
Kim'96 [47]	struct	1,800	free	4 PC	6 PC	93.13%
*Nakagawa'96 [88]	struct	3,345	fluent	N/A	11951 PW×30	80–90%
Chou'96 [18]	struct	5,401	flu-regular	5 PC	15 PC	93.4%
Wakahara'97 [134]	struct	2,980	careful/free	120 PC	36 PC	98.4/96.0%
Kim'97 [48]	HMM	1,800	free	4 PC	6 PC	90.3%
Zheng'97 [149]	FARG	3,755	regular	N/A	6 PC	98.8%
Xiao'97 [139]	struct	3,755	flu-regular	35 PC	3 PC	93.9
Nambu'98 [94]	struct	3,942	flu-regular	200 PC	200 PC	89.7%
Kuroda'99 [58]	statis	1,000	regular	25 PC	10 PC	94.34%
*Okamato'99 [99]	statis	3,345	fluent	11951 PW×40	11951 PW×41	86.32%
*Yasuda'99 [143]	HMM	3,057	fluent	10038 PW×100	10038 PW×20	85.89%
*Tanaka'99 [119]	combined	3,356	fluent	Nakayosi	Kuchibue	87.6%
Zheng'99 [150]	sta-struct	3,755	mixed	100 PC	7 PC	95.52%
*Akiyama'00 [1]	struct	3,345	fluent	6690 PW×150	11951 PW×3	88.58%
Shin'02 [113]	struct	2,965	regular	90 PC	24 PC	99.28%
*Kitadai'02 [53]	struct	3,345	fluent	9309 PW×163	11951 PW×120	87.2%
Tokuno'02 [121]	HMM	1,016	fluent	50,986	42,718	92.0%
Velek'02 [129]	combined	3,036	fluent	3,669,089	54,927	94.14%
Nakai'02 [92]	HMM	1,016	fluent	34 PC	34 PC	93.1%
Matic'02 [79]	neural	4,400	N/A	80 PC	20 PC	97.3%
Rowley'02 [106]	struct	6,847	natural	5 million	85,655	94.45%

PC: per category
PW: per writer
*Tested on TUAT Kuchibue database

Figure 1.1: Recognition rates reported in the literature

1.2.3 Development of Appropriate Evaluation Metrics

1.2.3.1 Choice of Evaluation Subjects

It is difficult to perform an accuracy evaluation of a recognition system that can be compared to other systems. The methods the systems use in order to perform their recognition are diverse. There is always a trade-off between robustness, performance and accuracy. The prototype that is subject to this evaluation performs a different task than most of the other recognition systems. It analyses the characters not only for the purpose of recognition, but attempts to create feedback on how well the input matched the character model in structural terms. That means concretely, the system can analyse an input with an expected result and can perform the same for an unknown input by assuming the best match as the expected result. The analysis yields an output that includes more information than pure pattern matching. It rather includes structural linguistic information.

For example, the characters 垉, 椛, 磆, 糀 and 錵 all share the substructure on the right: 花. Figure 1.2 shows the substructure on top and below five characters that are using the structure. The system analyses and distinguishes substructures. If the system is set up to recognise an input with an expected result the output contains a confidence value about the input quality concerning that character. Additionally, the error

Figure 1.2: Similar characters that can be confusing to learners: Five different Kanji share the same Radical in the *tsukuri* position (on the right).

recognition module returns information about the substructures that were found in the input. The similarity of the characters from figure 1.2 is known to the recognition system because of the identical substructures. If the system identifies the input as a character with identical parts with respect to the expected character the output will contain the information as a substructure confusion error type.

This detailed analysis creates a unique requirement for evaluation. Not only recognition accuracy must be measured and compared to other systems, but a new evaluation for the recognition of substructures is needed.

It seems reasonable to measure the recognition accuracy in a plain percentage of correctly recognised characters. Precision and recall are the correct measurement for the recognition of incorrect substructures. That way, true positives, false positives, true negatives and false negatives can be distinguished. Since the system can recognise pure substructures as well, it would be interesting to see the accuracy of a Radical recognition.

1.2.3.2 Metrics Details

The lexicon is a sample lexicon that contains 50 characters. That circumstance has to be kept in mind when comparing the evaluation results to other systems. Creating a lexicon entry for a single character is a labourious task. It would be outside the scope of the thesis to create a large lexicon. Three experiments will be conducted in order to evaluate the analytical handwriting recognition engine:

1. **Overall recognition accuracy**
2. **Error recognition accuracy**
3. **Substructure recognition accuracy**

The overall recognition accuracy (experiment 1) will be calculated as a measure of the n-best matches for a character input. N is defined as the lowest position in the list of n-best matches that will still be considered as a match of a character. That is, if N is set to 1, then only the best match will be considered. If $N = 3$, the best three matches will be taken into account. The overall recognition accuracy A_N is defined as the weighted percentage of correctly recognised characters when taking the N best matches account. The number of samples will be called S. n is an integer that marks the position in the list of n-best matches. The values for n that will be considered for the evaluation lie on the open interval $]1, N[$. If an input produces a confidence value for a character recognition high enough to be in a position where $n \leq N$, then $\frac{1}{n}$ will be added to the number of correctly recognised characters. That means, if a character is the best match, then $n = 1$. Therefore $\frac{1}{n} = \frac{1}{1} = 1$ will be added to the number of correct matches. If the correct character is the second best match, $\frac{1}{2}$ will be added. If the character has been recognised as the n-best match, but $n > N$ or the character has not been recognised at all, nothing will be added. f is a helper function that returns a value according to the relation between n and N. The variable n_i denotes the position of the i^{th} input character in the list of best matches for that character. n_i serves as an input value for f. A_N will be calculated as follows:

$$f(n, N) := \begin{cases} \frac{1}{n} & \text{if } n \leq N \\ 0 & \text{if } n > N \end{cases}$$

$$A_N := \frac{1}{S} \cdot \sum_{i=1}^{S} f(n_i)$$

An example for the application of that term would be to set $N = 1$. Only the best match will be considered for an input. If 100 input sequences are tested, then $S = 100$. Say, the input yields a high confidence value for the correct character, so that the character becomes the most salient in the list of best matches and occupies the first position. Then $f(1, 1)$ yields 1. Therefore, in the summation of the $f(n_i$ the value 1 will be added for each correctly recognised character. If 69 of the 100 input characters are correctly recognised, then $A_N = A_1 = \frac{1}{100} \cdot 69$.

That would equal a result of 69%. The key to this evaluation method is of course the value of N. Multiple experiments with different N-values will be conducted for best comparability.

Different possibilities exist for the definition of a baseline for that experiment. A baseline based on pure randomness would be $\frac{1}{C}$ with C representing the number of characters in the database. For $C = 50$ the baseline would then be $\frac{1}{C} = \frac{1}{50} = 0.02$. That is already a very low baseline. The larger the lexicon grows the lower the baseline for evaluation. For a lexicon that contains around 2,000 characters and covers the Jōyō Kanji the baseline would come down to $\frac{1}{C} = \frac{1}{2000} = 0.0005$. This baseline is certainly not sufficient to measure the quality of a recognition system. However, a significant baseline can be defined with the formula for the computation of A_N and generous assumptions about the accuracy of the recognition:
Assume, $N = 3$ and all the sample characters are among the first three matches in the list of matches, equally partitioned. Then one third would yield $n = 1$ with $f(n) = 1$, another third yield $n = 2$ with $f(n) = \frac{1}{2}$, and the last third yield $n = 3$ with $f(n) = \frac{1}{3}$. That leads to a baseline (B) calculation as follows:

$$
\begin{aligned}
B &:= \tfrac{1}{S} \cdot (\sum_{i=1}^{\frac{S}{3}} f(1,3) + \sum_{i=1}^{\frac{S}{3}} f(2,3) + \sum_{i=1}^{\frac{S}{3}} f(3,3)) \\
&= \tfrac{1}{S} \cdot (\sum_{i=1}^{\frac{S}{3}} 1 + \sum_{i=1}^{\frac{S}{3}} \tfrac{1}{2} + \sum_{i=1}^{\frac{S}{3}} \tfrac{1}{3}) \\
&= \tfrac{1}{S}(\tfrac{S}{3} + \tfrac{S}{3 \cdot 2} + \tfrac{S}{3 \cdot 3}) = \tfrac{1}{3} + \tfrac{1}{6} + \tfrac{1}{9} = \tfrac{11}{18} = 0.61
\end{aligned}
$$

It would be interesting to define

The error recognition accuracy (experiment 2) will

For the experiment concerning the overall recognition accuracy two writers wrote all 50 characters as input for the system. The overall will be calculated as the average of the The percentage of correctly recognised characters

Document created on Saturday 19th June, 2010 at 23:54