Using Semantics in Matching Cursive Chinese Handwritten Annotations

Matthew Y. Ma[1] and Patrick S.P. Wang[2], IAPR Fellow

[1] Panasonic Information and Networking Technologies Laboratory, Panasonic Technologies, Inc., 2 Research Way, Princeton, NJ 08540, USA
[2] College of Computer Science, Northeastern University, Boston, MA 02115, USA

Abstract. We propose a semantic matching network for the matching of cursive Chinese handwritten annotations. This architecture combines the semantics of Chinese language with the traditional elastic ink matching. Using semantics can make the matching algorithm more intelligent by pre-selecting the most likely candidates before elastic ink matching is applied thus speed up the whole matching process. The semantic matching network can also establish a link between Chinese handwritten annotations and typed text, which can be used to match between these two. Our experiments show that $75-85\%$ recall can be achieved with a speed improvement of 85% over traditional elastic ink matching.

1 Introduction

Pen computers and PDA's have been existing for more than a decade. The use of stylus in a pen computing system has already brought advantages in many ways, but handwriting recognition (HWX) proved to be a more difficult problem than most people first expected. Instead of HWX, some research has been done on electronic ink matching [5] that tries to match a query against raw electronic ink data without attempting to recognize them. Similar research can also be found in the works of [10] and [11], but no applications were identified. Pavlidis *et al* [9] has used shape metamorphosis to recognize on-line handwritten patterns, but their work is limited to handwritten patterns with small number of stroke segments mainly single words or simple shapes.

The traditional elastic matching incorporating a dynamic programming procedure was previously described by Tappert [12] on its applications in on-line handwriting recognition. A variation of elastic matching was proposed by Lopresti *et al.* [4] and it has been used on the matching of handwritten annotations based on stroke level. This algorithm can achieve high accuracy rate in spite of the variations of the way people write. This is specially appropriate for handling informal cursive Chinese handwriting. Yet it was quite slow.

While the traditional elastic matching algorithm matches handwriting without attempting to recognize it, a human, on the other hand, might take a different approach. As illustrated in Figure 1, when a human tries to match two pieces of handwriting, he/she first tries to identify their semantics, then matches them. In Chinese writing, radicals, which are small structural parts of a character, are basic elements of semantics. They usually have their own meanings.

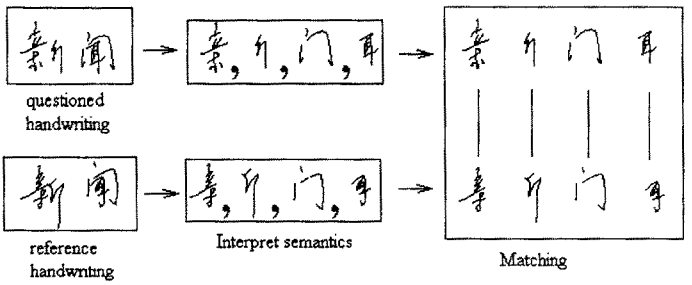

Fig. 1. Use of semantics when matching two pieces of handwriting in Chinese.

Based on this concept, we extended the elastic ink matching algorithm to be able to identify radicals in handwritten Chinese. These extracted radicals will then be used to "classify" the entries in the annotation database. When a query is entered by the user, only items with relevant meanings (radicals) in the database will be searched; therefore, the existing ink matching can be sped up by reducing the size of the problem. We can also convert handwritten Chinese annotations into a sequence of computer codes by this approach. This process does not involve a large vocabulary hence is simpler than traditional HWX. This scheme is suitable for matching ink annotations and it is called *semantic matching*.

The remainder of this paper is organized as follows. In Section 2 we describe our ongoing research on semantic matching and illustrate some interesting results. The performance of ink matching with and without using semantics is discussed in Section 3. Finally, in Section 4, we give our conclusions.

2 Semantic Matching

Our work focuses on the user-entered annotations in a document system. This task specially requires matching informal cursive handwriting. In this section, we discuss the concept of Chinese handwriting matching that uses semantics in the *Learning by Knowledge* paradigm. We illustrate that semantics can be used not only in the character recognition or linguistic processing, but also in the early process of ink matching. Our ongoing research on semantic matching focuses on two tasks: identifying radicals from handwriting to speed up the matching process and matching between handwriting and typed texts via radicals.

2.1 Semantic Matching Scheme

Learning by Knowledge is a new methodology presented by Wang [13], in which old knowledge can be accumulated and iteratively used to enforce and help to learn new knowledge via a feedback system. Figure 2 shows a learning cycle for Chinese character recognition process using this method.

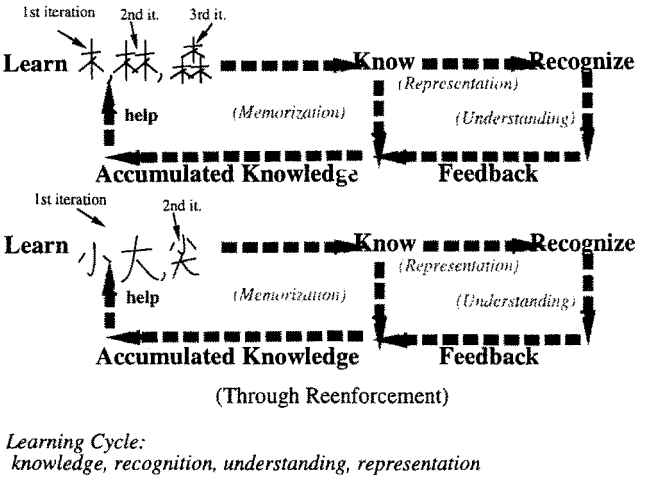

Fig. 2. Learning by knowledge in pattern recognition with examples.

Based on this concept, we constructed a semantic matching network as illustrated in Figure 3. In this architecture, our semantic knowledge base was obtained from a learning process (training) and can then be applied in the early processing stage. This approach helps to eliminate unwanted candidates via classification, which in turn speeds up the recognition (matching) process. The knowledge learned from semantics may also be used to convert handwriting to computer coded representation of radicals thus make the handwriting searchable by typed text query.

Our method of "semantic matching" and "learning by knowledge" has advantages over others in the literature. For example, the "learning by rote" method [1] adds new information to the process' learning structure without establishing any relationship with the existing concepts. The "meaningful learning" method [7] roughly relates new information to the relevant concepts already existing in the knowledge structure of the learner.

In the remainder of this section, several details of semantic matching will be described. Pre-processing and character segmentation were described in our previous work [6]. Similar work can also be found elsewhere [12][8].

2.2 Radical Extraction

This process identifies the radicals that are contained in each handwritten character. Almost any Chinese HWX systems have to deal with extraction of radicals or similar information. Some systems [15][16][3] extract feature stroke segments and then form them into radicals. This method would most likely fail on cursive Chinese handwriting since there are so many variations in its writing style. Some OCR systems extract radicals based on an assumption that there exist

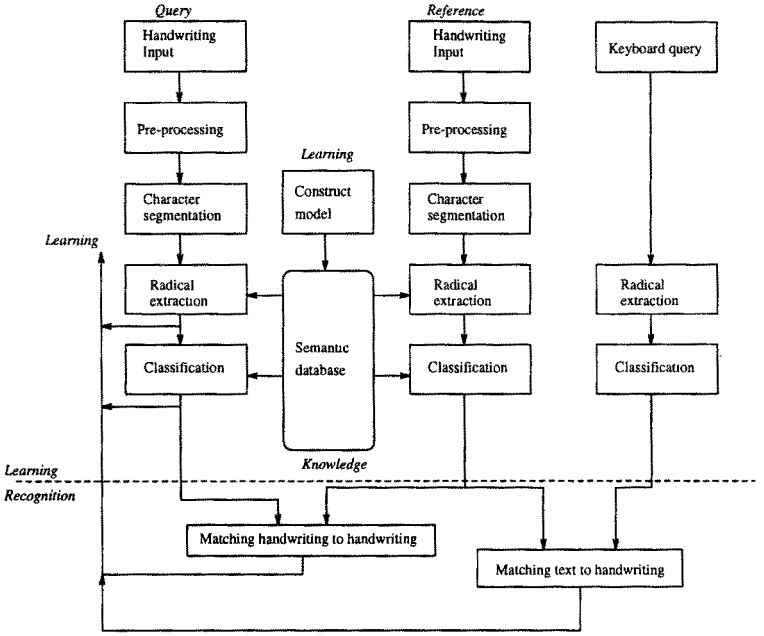

Fig. 3. Diagram of semantic matching.

gaps between radicals. This again can not apply to cursive Chinese handwriting because radicals are often connected together within a character.

On the other hand, the temporal order of strokes is often preserved in on-line handwriting. A Chinese writer often writes with a fixed stroke order. This order is taught when a child learns how to write. As personal writing style changes, this temporal order may change. But one person normally writes in a consistent way. This applies to radicals as well.

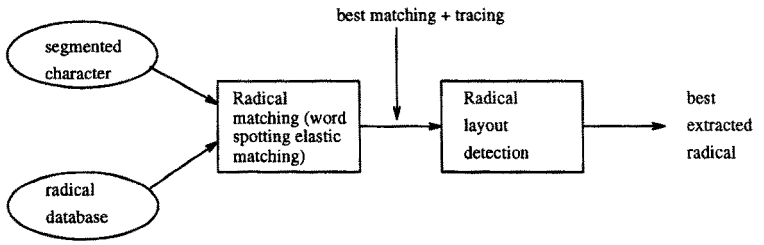

Fig. 4. The radical extraction scheme.

In our radical extraction scheme (illustrated in Figure 4), we combine the word spotting (or partial matching in [5]) version of elastic matching algorithm

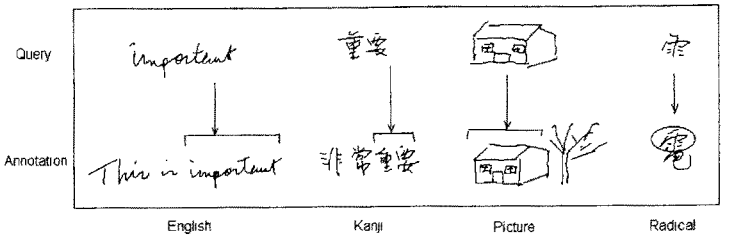

Fig. 5. "Word" spotting (subsequence matching).

with layout detection. The concept of word spotting has a number of applications as shown in Figure 5, including matching a radical to a character.

The semantic database (also called radical database) consists of a set of basic radicals, which are generated through a training process. The number of samples needed is small since the number of basic radicals in Chinese is very small (200 or so) in comparison to the large number of Chinese characters (3000 or so) that are commonly used.

The layout detection of radicals is based on the fact that for a particular radical, it can only reside at a known location within a character. For instance, Cheng and Hsu[2] classify radicals into seven layouts. In order to mathematically represent and compare these layouts, we define the *radical layout profile* as a vector of 9 elements. As illustrated in Figure 6, the value of each element is the number of stroke points that reside in each region.

When the semantic database is generated, the layout profile for each radical can be pre-computed in the training process. In implementation, every input character is matched partially with limited number of radicals in the semantic database. The top ranked radicals are then traced back to the original input character in order to compute the layout profile for the radical candidate. The similarity between this newly computed layout profile and the pre-computed profile for the extracting radical is then calculated. Let $P_R = \{r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9\}$ be the normalized layout profile (by its maximum value) for a radical in the semantic database, and $P_T = \{t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9\}$ be the normalized profile for

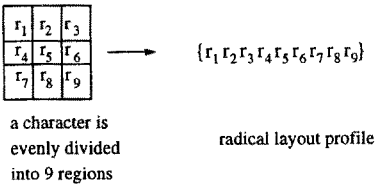

Fig. 6. The definition of radical layout profile.

traced radical stroke. The similarity measure is defined as:

$$S = c(P_R, P_T) - d(P_R, P_T), \tag{1}$$

whereas $c(P_R, P_T)$ is the correlation expressed as:

$$c(P_R, P_T) = \sum_{i=1}^{9} r_i t_i \tag{2}$$

and $d(P_R, P_T)$ is the distance between the two, thus is defined as:

$$d(P_R, P_T) = \sum_{i=1}^{9} (r_i - t_i)^2. \tag{3}$$

The similarity values are used to fine tune the order of top ranked extracted radicals. When the distance is greater than the correlation, the similarity measure will be a negtive value, and this value will be ignored in sorting the ranks. Since the number of radicals in the semantic database is limited, the matching time will be small. Figure 7 illustrates an example of a radical layout profile and its similarity to the profile for a radical model.

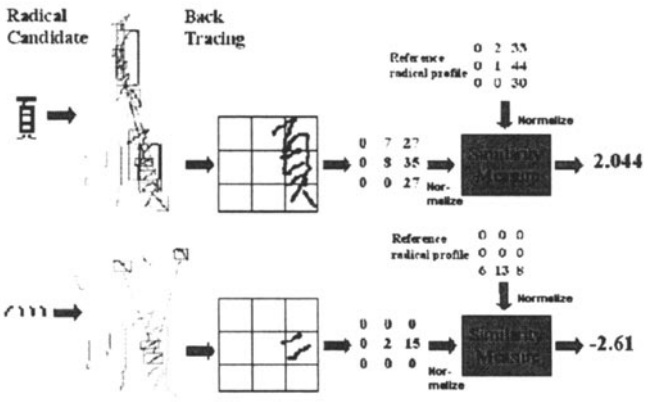

Fig. 7. An example of radical detection, radical layout profile and similarity measure.

As shown in Figure 7, when radicals in the semantic database are matched against a questioned character, the elastic matching with word spotting tries to find the best location that the radical fits within the character. This location can be traced back in the ink matching algorithm in order to compute the radical layout profile. With the similarity measure, the correctly located radicals yields a higher value than those not. It is shown in our earlier work [4] that the performance of radical extraction significantly improves after the layout detection is being used.

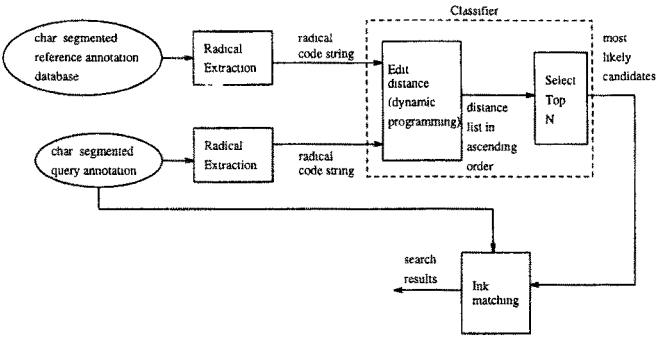

Fig. 8. Classification scheme.

2.3 Classification and Matching

In classification procedure, as shown in Figure 8, we use the results from radical extraction to obtain the most likely candidates for matching. First, each character is represented by $0-2$ numbers that correspond to extracted radicals. Each handwritten annotation is then represented by a sequence of radical numbers. Figure 9 shows the radical code representation of a sentence excerpted from a poem by Chinese famous poet Li Po.

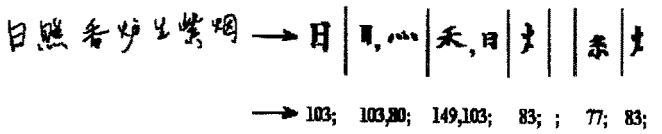

Fig. 9. Radical representation of annotations.

For each annotation in the reference database, the edit distance between its radical code sequence and that of a given query is calculated. The annotations with smaller distances can be selected as most likely candidates. To compute the edit distances for radical codes, again, we use the dynamic programming procedure. In this procedure, costs of insertion and deletion were assigned constant numbers. The cost for substituting one character with another is determined by the number of unique radical codes in the combination of the two. Consider the following two characters in the radical code representation: $c_1 = (1, 3)$ and $c_2 = (3, 17)$, the substitution cost is 2. As can be seen, currently all the radicals are assigned equally weight in computing the cost.

To make the matching more reliable, each annotation in the most likely candidate list will be matched against the query annotation using the same elastic ink matching as described earlier in this paper, and a top match list is generated. As illustrated in Figure 3, the matching can be applied to both ink vs. ink query and ink vs. typed text query.

3 Experimental Evaluations

We examined the following two issues: the effect of information reduction by the use of the semantics on the classification process, and the trade-off between the speed improvement and recall rate if any.

In our experiments, 200 Chinese annotations (translated movie titles in Chinese) were randomly selected. On an average, the annotations are 5 characters long. Four subjects were asked to write these annotations twice, one set as reference and the other as query. Each annotation was then converted into the corresponding radical code representation. The training process requires each subject to write only 45 characters, each containing one of the 45 radicals identified by Xiao and Dai[14] as illustrated in Figure 10. From these samples, radicals were extracted manually to construct a semantic database for each user, and the similarity measure for each radical was pre-computed as well.

Fig. 10. The 45 radicals identified by Xiao and Dai and the samples of radical training.

For each query, we selected a list of candidates from the reference ink database based on the edit distance of radical codes. We then applied elastic ink matching on the selected candidates and constructed a top match list according to their matching values. Figure 11(a) shows the results for the number of selected candidates (NoC) to be 30. The recall performance for semantic matching is $7-8\%$ lower than that of traditional matching reported in [4]. On the other hand, the search time was reduced to approximately $\frac{1}{7}$.

In theory, the recall performance improves as the NoC increases. This, however, would increase the computation time since more selected candidates (annotations) will be used for matching. In reality, a good trade-off value of NoC has to be obtained for optimal performance.

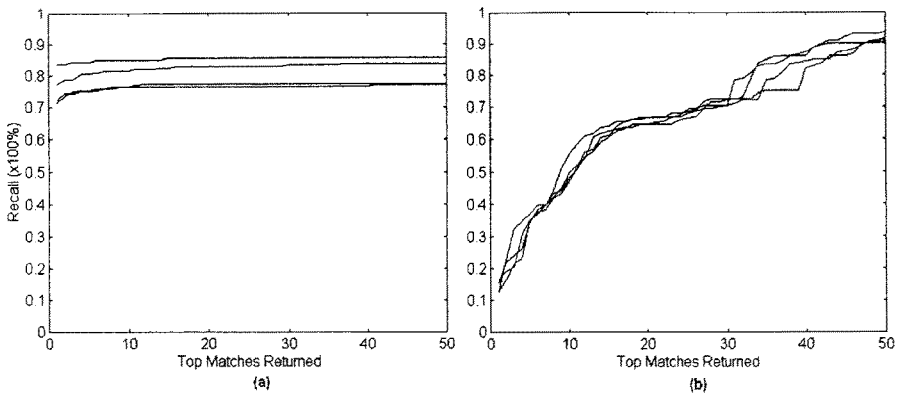

Fig. 11. Semantic matching: Recall vs. number of top matches returned for 4 subjects. (a) Matching handwriting with handwriting at $NoC = 30$; (b) Matching typed text queries with handwritten annotations.

Figure 11(b) shows the recall rates by typed text queries. The recall at the first hit is only 15% or so, however, the recall climbs up very quickly and reached about 65% with 20 (accounts for 10% of the size of the database) top matches returned. This gives us an indication that the semantic matching scheme is likely to work for searching handwritten annotations by typed text queries but needs some improvements.

4 Conclusions and Future Research

In this paper, we proposed a semantic matching scheme for matching cursive Chinese handwritten annotations. The experiments show some promising results on using semantics in the matching of Chinese handwritten annotations. By using semantics, significant improvement over speed can be achieved while maintaining reasonable high retrieval rate. Another potential of our approach is that it needs only a small training set, currently as little as 45 handwritten characters are required for each user.

We are also relatively in the early stage of our research on using semantics to match between handwritten annotations and typed text. Our preliminary experiments demonstrated the feasibility of this method. To enhance its performance, the underlying radical extraction has to be improved.

There are several ways to improve the radical extraction. First, the optimal number of radicals and the selection of radicals may be obtained from an empirical training process. Secondly, careful selection of radicals that is more immune to noise can increase the performance of radical extraction. Third, in our current classification procedure, all radicals were equally treated in calculating the costs for dynamic programming procedure. Different scaling factors for these costs may be assigned to different radicals depending on their similarity.

The ink matching scheme we proposed is more sophisticated than the conventional "syntactic" methods without contextual information [13][7]; it requires more memory and a backtracking procedure. More work need to be done in the future to overcome these difficulties. A larger dictionary (lexicon) and more intelligent machine of inferring and reasoning will also be helpful.

References

1. D.P. Ausubel, J.D. Novak, and H. Hanesian. *Educational Psychology: A cognitive view.* New York: HRW Co., 2nd Ed., 1978.
2. F. Cheng and W. Hsu. Research on Chinese OCR in Taiwan. In *Character and Handwriting Recognition. P.S.P. Wang (ed.)*, pages 139–164. World Scientific, 1991.
3. J. Liu, W.K. Cham, and M.M.Y. Chang. Stroke order and stroke number free on-line Chinese character recognition using attributed relational graph matching. In *Proc. 13th ICPR*, pages 259–263, August 1996.
4. D. Lopresti, M. Ma, P.S.P. Wang, and J. Crisman. Ink matching of cursive Chinese handwritten annotations. *Int. J. of PRAI*, 12(1):119–141, 1998.
5. D. Lopresti and A. Tomkins. On the searchability of electronic ink. In *Proc. of the 4th Int. Workshop on Frontiers in Handwriting Recognition*, pages 156–165, 1994.
6. M. Ma, P.S.P. Wang, D. Lopresti, and J. Crisman. Semantic matching of free-format Chinese handwriting. In *Proc. of the 17th Int. Conf. on Comp. Proc. of Oriental Lang.*, pages 107–111, Hong Kong, April 1997.
7. R.B. Millward. Models of concept formation. In *Aptitude, Learning, and Instruction. R.E. Snow et al (eds.)*. L. Erlbaum Assoc. Hillsdale, NJ, 1980.
8. H. Murase. On-line recognition system for free-format handwritten Japanese characters. In *Character and Handwriting Recognition. P.S.P. Wang (ed.)*, pages 207–220. World Scientific, 1991.
9. I. Pavlidis, R. Singh, and N.P. Papanikolopoulos. An on-line handwritten note recognition method using shape. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, pages 914–918, 1997.
10. A. Poon, K. Weber, and T. Cass. Scribbler: A tool for searching digital ink. In *Companion Proceedings of the CHI*, pages 252–253, 1995.
11. D. Reynolds, D. Gupta, and R. Hull. Architectures for efficient scribble matching. In *Proc. of the 4th Int. Workshop on Frontiers in Handwriting Recognition*, pages 488–495, 1994.
12. C.C. Tappert. Speed, accuracy, and flexibility trade-offs in on-line character recognition. In *Character and Handwriting Recognition. P.S.P. Wang (ed.)*, pages 79–96. World Scientific, 1991.
13. P.S.P. Wang. Learning, representation, understanding and recognition of words - an intelligent approach. In *Fundamentals in Handwriting Recognition. S. Impedovo (ed.)*, pages 81–112. Springer-Verlag, 1994.
14. X.-H. Xiao and R.-W Dai. On-line handwritten Chinese characters recognition directed by components with dynamic templates. In *Proc. of the 17th Int. Conf. on Comp. Proc. of Oriental Lang.*, pages 89–94, Hong Kong, April 1997.
15. S.L. Xie and M. Suk. On machine recognition of hand-printed Chinese characters by feature relaxation. *Pattern Recognition*, 21(1):1–7, 1988.
16. K. Yamamoto and H. Yamada. Recognition of handprinted Chinese characters and Japanese cursive syllabary. In *Proc. 7th ICPR*, pages 385–388, Montreal, 1984.