A FLEXIBLE TEMPLATE LANGUAGE MODEL AND ITS APPLICATION TO HANDWRITING RECOGNITION

Homayoon S.M. Beigi
T.J. Watson Research Center
International Business Machines
P.O. Box 704
Yorktown Heights, New York 10598

Abstract

Commonly, in handwriting recognition, once a character is written, its digitized $< x, y >$ coordinates are fed into a shape matcher which will match some features with a set of given prototypes and will return probabilities of each prototype being the intended writing. However, some applications in handwriting recognition arise which are very limited in their format or their vocaulary. Examples of these applications are the cases when only numbers are allowed or when for a specific field could only be an upper-case character or when a format of 2 digits and one number is expected. Many other examples arise in which at any point, the fanout of possibilities for a particular location is greatly reduced from the normal case of all symbols being enabled. Reducing the fanout will reduce the number of prototypes that the shape matcher should match against. [1] This will reduce the number of confusions and will also reduce the search space which will result in much better accuracies. In fact reducing the number of prototypes to about 1/3 has shown to reduce the error by about 70%. [1] This paper presents a robust template language model which affects the search to increase the accuracy of the handwriting recognizer in the cases when a reduction in the search fanout is possible.

1 Introduction

The notion of using a handwriting interface to computers has been of great interest to many researchers for a number of years. With today's technology, a low-priced LCD digitizer tablet for capturing handwriting is conceivable with a size no bigger than a notebook. These devices capture the $< x, y >$ coordinates of the writer's pen on the surface of the digitizer tablet. Also, with the introduction of new pen-based operating systems such as the Penpoint^{TM} and Pen-Windows^{TM} operating systems handwriting recognition applications have become very practical. For a survey of different handwriting recognition methods see [2].

In many application of handwriting recognition, such as form filling applications, the information written in a specific writing field is constrained. These constraints could be the imposition of a limit on the alphabet, some specific writing format or the restriction to a small vocabulary. For some examples of the above take the following respectively:

1. A field containing dollar amounts which includes digits, '.' or '$'.

2. A field containing license plates. ("ccc-ddd" where 'c' stands for a member of the English alphabet and 'd' stands for a digit)

3. A login field for the computer which could only take names of people with an account in that computer.

For the above applications, a template is used which will restrict the candidates of recognition and therefore reduce the rate of confusion. Reducing the confusion will in turn increase the accuracy of the handwriting recognizer. [1] In fact reducing the number of prototypes to about 1/3 has shown to reduce the error by about 70%. [1]

As one solution to reducing the fanout at any point in recognition, one could disallow any characters which are not allowed by the field. However, this will restrict the writer to abide by the rules set by the template and will not leave any room for exceptions. Therefore, if for example someone happens to have a custom license plate with his name on it, he would never be able to write it in that field and to take care of these exceptions, then the specific format could not be used to the advantage of increasing the recognizer accuracy.

This paper introduces a flexible template language model which affects the search by reordering and rescoring the search hypotheses to give a higher precedance to those hypotheses which fall within the restrictions of the specific template. However, this will still leave room for the possible recognition of hypotheses which do not meet the template restriction. The proposed language model will therefore, take advantage of the template format in a probabilistic fashion for increasing the accuracy of recognition and lets carefully written statements still be recognized even if they do not meet the template restrictions.

This language model has been implemented in a stroke-based handwriting recognizer which uses elastic matching[1] for its shape matcher. Another feature of this language model is its ability to add stroke hypotheses to the list of stroke-match hypotheses. These hypotheses are added to fill in for poorly matched strokes.

At any point in the search, the context of the recognized portion of the writing is passed to a template module which will provide all the possible characters following that context. Then, the predictions of this module are used for reordering the priorities of search and rescoring the hypotheses and even adding to the list of hypotheses. This will in general include any of the above 3 template types.

2 Description

Due to the nature of the elastic-matcher employed in the recognition system [1] and the heuristics involved, the scoring system is not perfectly probabilistic. However, the template language model could produce a probabilistic set of scores (even if they might be based on a uniform distribution). One of the biggest problems is to develop a method for mixing the scores provided by the recognizer and those provided by the template model. An array of runs were made using the recognition engine without the template language model, to collect statistical data about the heuristic scores and their mapping to probabilities. As a correct answer came back from the recognizer, the recognition scores for the strokes in that word were noted. After finishing the recognition of the handwriting of over 24 writers each writing 180 words, the percentages of the successful words were noted with their corresponding recognition scores. These values were then integrated (sumed) to give an accumulated value such that the best possible score would map into a probability of 1. Figure 1 shows a graph mapping recognition scores to probabilities.

Figure 1: Score-Probability Mapping

The information in figure 1 is used in the form of a look-up table to convert between the recognition system scores and probabilities. Figure 2 shows the details of the list of hypotheses attached to a search node as it is created. In the absence of the template language model, each search generation (corresponding to a stroke) is provided with a list of stroke hypotheses with stroke-match scores by the matcher. When the template language model is used, each node, depending on its language model path could have a different list of stroke hypotheses. Some strokes based on the language model path of the node, are not allowed by the template model or they have template language model scores which might not compete with their stroke-match scores. Therefore, as in figure 2, there is a list of local stroke hypotheses associated with every new node which is generated through search. This list includes in each entry, an index to its position in the global hypothesis list for that generation. In addition, it includes a score which is a mixture of the language model probability mapped into score and the score given by the basic recognition engine.

Figure 3 shows a flow-chart of the proposed template language model. Based on this figure, as a new search node is generated, it is checked for being a last stroke within a character hypothesis. Each stroke is identified as in figure 2, by its position within a character (starting with 0) and the total number of strokes in that chracter. In addition, the stroke has a character prototype I.D. which is unique. If the search node is not the last stroke of a hypothesized prototype, no action is done and the search engine uses the global hypothesis list for continuing the search process.

If the search node is the last stroke of a hypothesized charcater prototype, it is sent to the template language model to be provided with a list of language model (LM) hypotheses with their corresponding probabilities of occurance after that search node. These template model probabilities are weighed against their context length. Namely, the longer the context at the search node, the more the template model probability is weighed. This provides some smoothing for the effects of the template language model.

As the next step, all the stroke hypotheses which have both a poor match score and a poor language model score are killed. At this point the language model probabilities are mapped into scores which the search operates in, through the mapping information available in figure 1.

At this point a new score is created throught the mixture of the stroke-match and template model scores. This mixture is a weighted sum of the two scores. The weighting factor partly governs the strength of the language model used.

Once these scores are generated, the list of hypotheses is sorted based on the newly generated scores and then the number of strokes in the character prototypes in ascending order of number of strokes. This new list of hypotheses is then used by the search algorithm to generate new children for that node.

3 Test and Conclusion

The template language model was tested and in most cases it increased the accuracy close to 99-100%. This language model has basically been designed such that exceptions are allowed. An example of these exceptions is when a motor vehicle license plate is set to have a ccc-ddd format where c stands for character and d stands for digit. In a case like this one could disallow any sequence other than ccc-ddd. However, the basic idea against the proposed template model is that it will allow a well-written exception to be accepted even if it does not abide by the format imposed by the template. An example of such exceptions is when a person has a special license plate say with his name on it.

References

[1] T. Fujisaki, H.S.M. Beigi, C.C. Tappert, M. Ukelson and C.G.Wolf, "Online Recognitoin of Unconstrained Handprinting: A Stroke-based System and Its Evaluation," _______, S. Impedovo, Editor, ELSEVIER: Italy, 1992.

[2] C.C. Tappert, C.Y. Suen and T. Wakahara, "The State of the Art in On-Line Hand-

writing Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, N0. 8, pp. 787-808, August 1990.

Figure 2: Local Search Hypothesis List

Figure 3: Flow-Chart of the Template Language Model