Combination of Multiple Handwritten Text Line Recognition Systems with a Recursive Approach

Roman Bertolami, Beat Halter, Horst Bunke

Combination of Multiple Handwritten Text Line Recognition Systems with a Recursive Approach

Roman Bertolami, Beat Halter, Horst Bunke
Institute of Computer Science and Applied Mathematics
University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
{bertolam, bhalter, bunke}@iam.unibe.ch

Abstract

In this paper we propose a novel method to combine the results of multiple text line recognition systems. The method uses a recursive approach and re-examines those parts in a text line which have been rejected based on the initial combination of the base recognisers' results. By means of the new method, the search space can be reduced, and therefore more accurate recognition results can be expected. Experiments conducted on the IAM database show that the proposed method is able to improve the recognition rate compared to a standard combination scheme.

Keywords: Handwritten Text Line Recognition, Multiple Classifier Systems, Hidden Markov Models

1. Introduction

Handwriting recognition has been addressed by many researchers. In character, digit, and isolated word recognition high recognition rates have been achieved which enable successful applications in the fields of postal address reading [3] and cheque processing [7]. Most of the systems reported in literature consider constrained recognition problems, involving a small vocabulary or specific writing instruments. The recognition of general handwritten text is still a widely unexplored field with many open problems. Everyone has their own writing style, different writing instruments can be used, and the number of word classes is usually huge. Furthermore, the difficult problem of segmentation occurs when moving from word to text recognition because the correct number of words in a text line is unknown in advance. Therefore, rather low recognition rates of only 50% to 80% have been reported in literature for general handwritten text recognition [9, 17, 19, 20].

Multiple classifier systems have successfully been applied to improve the classification accuracy in many different fields of pattern recognition [10, 14]. Voting and similar strategies have shown good potential to improve the classification accuracy compared to a single classification system. In the domain of handwriting recognition, classifier combination has often been applied for isolated character and single word recognition. However, the combination of handwritten text recognisers has been proposed only recently. This kind of combination requires some additional synchronisation mechanism because the number of words in the recognised word sequences might differ. Usually, a sequence alignment procedure is applied to synchronise the word sequences output by the individual base recognisers. Then, a standard voting method can be applied to extract the final result from the synchronised sequences.

The contribution of the present paper is a novel architecture for the combination of multiple recognisers, which involves recursive recognition. In the overall combined system, all parts of a text line are rejected where too many individual recognisers disagree. The rejected parts are subjected to an additional round of recognition. Rejection and re-recognition can be iterated several times. With an increasing number of rejection and re-recognition cycles, those parts of a text line that undergo recognition are becoming increasingly smaller. Hence the overall search space of the recogniser is gradually reduced. Because the search space is reduced, larger parts can be explored by the sub-optimal search strategy during the decoding step. In general, this leads to different recognition results. The probability that the correct word sequence is among the explored parts increases because a more exhaustive search is performed. Therefore, we can expect that the recognition becomes more accurate.

The remaining part of this paper is organised as follows. Related work is summarised in the next section. In Sect. 3 the recursive recognition procedure is introduced. Recognition, combination, rejection, cutting, and pasting, which are the essential steps of the proposed method, are explained in detail. Experiments and results are provided in Sect. 4 and conclusions are drawn in the last section of the paper.

2. Related Work

In offline handwriting recognition, improvements by means of multiple classifier systems have been reported for handwritten character, numeral, word, and word sequence recognition.

An automatic self-configuration scheme to combine multiple character recognition systems has been proposed in [18]. For this scheme genetic algorithms are used.

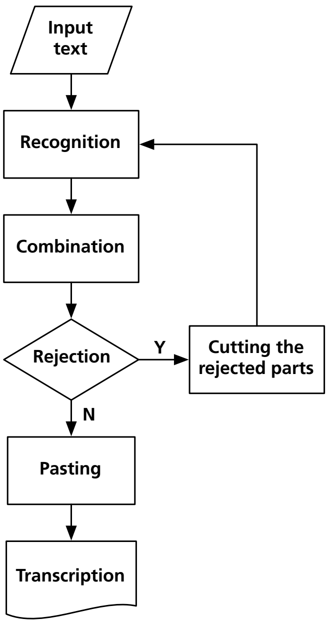

Figure 1. Recursive recognition overview.

In numeral recognition, the application of statistical combination methods has been reported in [6]. Especially the behaviour knowledge space methods were able to successfully combine the classifiers. In [22] a framework to combine numeral string recognisers was proposed. This framework uses a graph-based approach for combination.

An evaluation of several decision combination strategies for handwritten word recognition has been reported in [4]. Borda count methods, fuzzy integrals, and multilayer perceptrons have been compared. In [5] various ensemble methods, including bagging, boosting, and feature subspace methods have been applied to handwritten word recognition. Hidden Markov Model based recognisers have been automatically generated by modification of the training set.

In handwritten text line recognition additional effort is required to synchronise the word sequences. In [11] a heuristic approach to align and combine multiple handwritten text line recognisers has been used. Positional information of the recognised words is exploited to reduce the search space of the alignment. A novel method to generate ensembles of text line recognition systems has been introduced in [2]. Based on specific integration of a statistical language model multiple recognisers have been built.

3. Methodology

This section describes the proposed recursive recognition schema. First, an overview is given. The individual parts of the process are then described in greater detail. Figure 2 provides an illustrating guiding example of the entire recursive recognition process.

3.1. Overview

A system overview is shown in Fig. 1. First the handwritten input text is recognised by n independent base

(a) Recognition results of R_1, R_2, R_3:

R_1: leave is the autumn
R_2: leave in that autumn
R_3: leave is that autumn

(b) Alignment of the word sequences:

R_1:	leave	is	the	autumn
R_2:	leave	in	that	autumn
R_3:	leave	is	that	autumn

(c) Result of the re-recognition step:

R_1:	in	the
R_2:	in	the
R_3:	in	that

(d) Final result with the TakeRevoted pasting strategy:

leave in the autumn

(e) Final result with the TakeOriginal pasting strategy:

leave in that autumn

Figure 2. Example of the recursive recognition step.

recognisers. The results of these recognisers are then combined. Based on a confidence measure we reject certain parts of the combined result. The rejected parts are cut from the original input image and resubmitted to the recognition process. This recursion can be applied multiple times. Finally, the recognised parts are pasted together to build the final transcription.

3.2. Recognition and Combination

In the first processing step, the handwritten input text is recognised by n different base recognisers individually. The output of the recognisers are n word sequences each of which is a textual representations of the handwritten text line. Note that the number of words in these sequences may differ. In the example of Fig. 2 three base recognisers (R_1, R_2, R_3) output a transcription for the handwritten input text *leave in the autumn* (Fig. 2a). Notice that none of the base recognisers correctly recognises the input.

To combine the output word sequences of the base recognisers we first use an alignment procedure to synchronise the sequences. Then, we apply a voting strategy to the individual segments of the alignment.

A heuristic extension of the standard sequence alignment method [21] is used to align the word sequences. The heuristics enable us to reduce the search space by using positional information as proposed in [11]. The result of the alignment procedure is a sequence of segments which contain the recognised words. In our example the alignment results in four segments, as shown in Fig. 2b.

Table 1. Recognition rates of the three base recognisers.

Recogniser	Recognition Rate
R_1	63.06%
R_2	58.71%
R_3	55.33%

Table 2. Recognition rates of the different cutting (Plain/Average) and pasting strategies (TakeRevoted/TakeOriginal)

Baseline System		63.85%
Plain	TakeRevoted	63.78%
	TakeOriginal	64.59%
Average	TakeRevoted	63.19 %
	TakeOriginal	64.69%

To each of the aligned segments we then apply a weighted voting strategy to extract the combination result. The weights are proportional to the recognition rates of the individual recognisers.

3.3. Rejection and Cutting Methods

Depending on how many recognisers agree on their decision we define a confidence measure. Based on this confidence measure we can then reject certain parts of the input. For example, if each recogniser outputs a different word, we reject this part. In the example of Fig. 2 we only accept if all recognisers agree on their decision. Therefore, the second and the third word are rejected (Fig. 2b).

An additional round of recognition is then applied to the rejected parts. To be able to re-recognise the rejected parts we need a suitable cutting method. Because the starting and ending points of the recognised words may be different for the n recognised word sequences, it is not trivial to find suitable starting and ending points for the re-recognition step. If we choose the starting point of a part too far to the left, we risk to have an additional, already correctly recognised, word in the cut part. On the other hand, if the start point is too much to the right, part of a word can be removed, thus making a correct recognition impossible. The same problems but in opposite order occur for the ending point. For these reasons we propose two different cutting methods:

Plain The rejected parts are cut for each of the n recognisers individually. Thus, for each extracted segment we have n starting points $(s_1, \ldots, s_n)$ and n ending points $(e_1, \ldots, e_n)$. Recogniser R_i then performs the re-recognition on the part between s_i and e_i.

Average A common part is cut for all the n recognisers. The average starting point $s = \frac{1}{n}\sum_{i=1}^{n} s_i$ and the average ending point $e = \frac{1}{n}\sum_{i=1}^{n} e_i$ are used for this purpose.

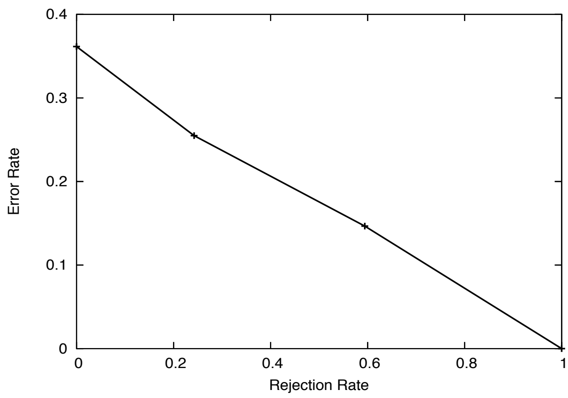

Figure 3. Error-rejection plot.

3.4. Recursive Recognition

Once we have cut the parts that need further examination, we can apply the re-recognition step. We resubmit the extracted parts to the same base recognisers. The motivation for this procedure is that we can dramatically reduce the search space when we don't have to recognise a whole text line but only a part of it. The re-recognition step is identical to the initial recognition except for the reduced input sequences. The same recognition procedure is used and the same statistical language model supports the recognition process. Nevertheless, we can expect more accurate recognition, because the decoding step which performs the recognition implements a sub-optimal search strategy. The search space of the re-recognised parts is usually much smaller than the original one. Because the search space is smaller it can then be explored more deeply. Thus, we increase the probability that the correct words are considered during the decoding step and can therefore expect recognition becoming more accurate. The result of the re-recognition step in our example is shown in Fig. 2c. While the recognisers now agree on first word *in*, they still disagree on the second word.

The results of the re-recognition steps are then combined according to the same combination scheme as the original recognition results. If there are still parts to be rejected we can recursively invoke another round of re-recognition. To stop the recursion process we simply define a maximal number of iterations.

3.5. Pasting Methods

Once the recursive recognition has finished we have to include the results of the cut parts in the original result. It may occur that some parts still do not fulfil the acceptance criteria. To get the final result for these parts as well we propose two different strategies:

TakeRevoted The results of the last recognition step are combined and provide the results for the cut parts.

TakeOriginal The results of the first recognition step are combined and provide the results for the cut parts.

In the example of Fig. 2, we decide to use the re-recognition step only once and therefore apply the past-

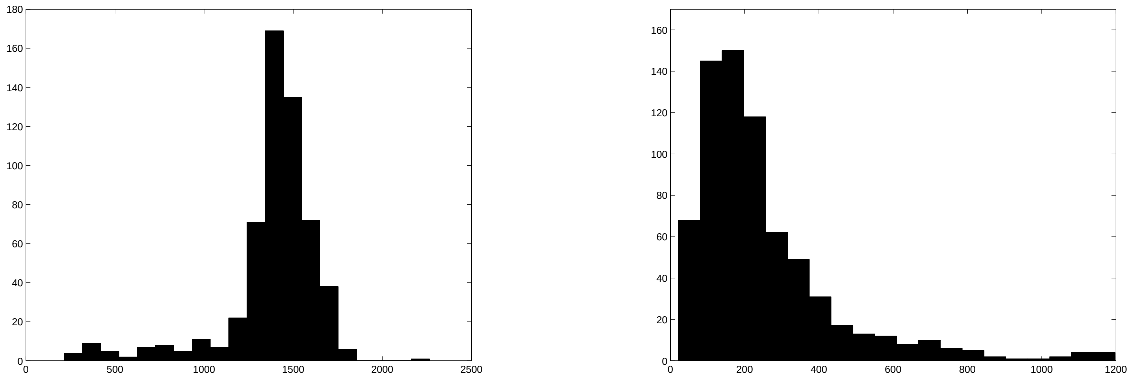

Figure 4. Distribution of the length of the input sequences in pixel. On the left, the lengths of the original text lines are shown whereas the lengths of the extracted parts which are submitted to re-recognition are shown at the right.

ing methods immediately after the result in Fig. 2c has been produced. The TakeRevoted method (d) provides the correct transcription whereas the TakeOriginal method (e) produces an error at the third word.

4. Experiments and Results

In the experiments we use three different base recognisers (R_1, R_2, R_3) which are combined according to the proposed scheme.

4.1. Experimental Setup

Each of the three recognisers (R_1, R_2, R_3) is based on Hidden Markov Models (HMMs) [15] and is trained on the same dataset. Additionally, the same statistical language model supports the decoding step. However, the recognisers are different in terms of feature extraction and state modeling.

Geometric features are used by recogniser R_1 [12]. For each character, an individual number of states is determined with the quantile method [23]. Six Gaussians are used to model the output distribution of each state.

The input of recogniser R_2 is a pixel-based feature stream [19]. The Bakis method is used to determine the number of states per character [1]. The output distribution is again modelled with six Gaussians.

The last recogniser (R_3) we use is also based on the geometric features introduced in [12]. In contrast to R_1 the number of states is determined with the Bakis method and only a single Gaussian is used to model the output distribution.

All data used to train and test the system originate from the IAM[1] database [13]. The HMM models are trained on 1530 text lines written by 35 writers. The test set consists of 572 text lines written by fifteen writers. The considered task is writer independent which means that no writer who contributed to the test set is used to train the system. The underlying lexicon contains 4207 word instances and is the union of all words occurring in the training and test set. The statistical language model we used to support the decoding step is a bigram language model [16] which was extracted from the LOB corpus [8].

[1]The IAM database is publicly available for download at http://www.iam.unibe.ch/∼fki/iamDB

4.2. Testset Results

The recognition rates of the three base recognisers (R_1, R_2, R_3) are summarised in Tab. 1. The geometric feature based recogniser R_1, which uses a mixture of six Gaussians, clearly outperforms recognisers R_2 and R_3.

As a baseline system we use the combination of R_1, R_2, and R_3 according to the combination methods described in Sect. 3 without rejections and re-recognition. The recognition rate of this baseline system is 63.85%. Thus, the combination of the three recognisers without recursion already outperforms the best single recogniser R_1. The goal is now to show that the recursive recognition has the potential to perform even better.

The error-reject characteristic of the new system is shown in Fig. 3. We can either force the system to accept each combination result (error rate: 36.15%), accept if at least two of the recognisers produce the same result (error rate: 25.49%), or accept only if all recognisers agree (error rate: 14.66%).

For the sake of simplicity we apply the recursive recognition only once. This means that the rejected parts are resubmitted to the recognition process only one time. We accept words that occur at least in the result of two recognisers. The other words are rejected and re-recognised.

The results on the test set are summarised in Tab. 2. We can see that the TakeOriginal pasting method is able to improve the performance whereas the TakeRevoted method leads to some performance decrease. The best performing system uses the Average cutting method and the TakeOriginal pasting method which yields a recognition rate of 64.69%.

One distinctive feature of the proposed method is a reduction of the length of the input image. This enables us to perform a more accurate recognition. The reduction of the length of the input sequences is illustrated in Fig. 4. In the recursive recognition step the input images are, on average, more than five times shorter than the original input.

5. Conclusions

We have proposed a novel method to combine multiple text line recognisers. The novelty we introduce is a recursive recognition step which enables us to resubmit the difficult parts of a text line to a second round of recognition. Because only parts of the text lines are re-recognised we can reduce the search space and therefore expect to obtain a more accurate recognition.

In the proposed system, each of the base recognisers outputs a transcription of the handwritten input text first. These word sequences are then combined. Based on how many recognisers agree in their decision, certain parts of the combination result are rejected. Next, we cut the rejected parts from the original image and resubmit them to the recognition process. All recognised parts are pasted together to build the final word sequence.

Experiments conducted on the IAM database show that the proposed method is able to improve the recognition rate compared to a standard combination scheme.

Acknowledgement

This research was supported by the Swiss National Science Foundation (Nr. 200020-19124/1). Additional funding was provided by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Visual/video processing".

References

[1] R. Bakis. Continuous speech recognition via centisecond acoustic states. In *Proc. of the 91. Meeting of the Acoustic Society of America, Washington, USA*, 1976.

[2] R. Bertolami and H. Bunke. Multiple handwritten text recognition systems derived from specific integration of a language model. In *8th International Conference on Document Analysis and Recognition, Seoul, Korea*, volume 1, pages 521–524, 2005.

[3] A. Brakensiek and G. Rigoll. Handwritten address recognition using hidden Markov models. In A. Dengel, M. Junker, and A. Weisbecker, editors, *Reading and Learning*, pages 103–122. Springer, 2004.

[4] P. Gader, M. Mohamed, and J. Keller. Fusion of handwritten word classifiers. *Pattern Recognition Letters*, 17:577–584, 1996.

[5] S. Günter and H. Bunke. Ensembles of classifiers for handwritten word recognition. *International Journal on Document Analysis and Recognition*, 5(4):224 – 232, 2003.

[6] T. Huang and C. Suen. A method of combining multiple experts for the recognition ofunconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:90–94, 1995.

[7] S. Impedovo, P. Wang, and H. Bunke, editors. *Automatic Bankcheck Processing*. World Scientific, Singapore, 1997.

[8] S. Johansson, E. Atwell, R. Garside, and G. Leech. *The Tagged LOB Corpus, User's Manual*. Norwegian Computing Center for the Humanities, Bergen, Norway, 1986.

[9] G. Kim, V. Govindaraju, and S. Srihari. Architecture for handwritten text recognition systems. In S.-W. Lee, editor, *Advances in Handwriting Recognition*, pages 163–172. World Scientific Publ. Co., 1999.

[10] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons Inc, 2004.

[11] U.-V. Marti and H. Bunke. Use of positional information in sequence alignment for multiple classifier combination. In *J. Kittler, F. Roli (eds.): Multiple Classifier Systems, MCS 2001, LNCS 2096, Springer*, pages 388 – 398, 2001.

[12] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[13] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39 – 46, 2002.

[14] N. Oza, R. Polikar, J. Kittler, and F. Roli, editors. *Multiple Classifier Systems, 6th International Workshop*. Springer LNCS 3541, 2005.

[15] L. Rabiner. A tutorial on hidden Markov models and selected application in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[16] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proc. of the IEEE*, 88:1270–1278, 2000.

[17] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

[18] K. Sirlantzkis, M. Fairhurst, and M. Hoque. Genetic algorithm for multiple classifier configuration: A case study in character recognition. In J. Kittler and F. Roli, editors, *2nd International Workshop on Multiple Classifier Systems (MCS), Cambridge, England*, pages 99–108, 2001.

[19] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720, 2004.

[20] A. Vinciarelli and J. Luettin. Off-line cursive script recognition based on continuous density HMM. In *7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands*, pages 493–498, 2000.

[21] R. Wagner and M. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[22] X. Ye, M. Cheriet, and C. Y. Suen. Strcombo: combination of string recognizers. *Pattern Recognition Letters*, 23:381–394, 2002.

[23] M. Zimmermann and H. Bunke. Hidden Markov model length optimization for handwriting recognition systems. In *8th International Workshop on Frontiers in Handwriting Recognition, Niagara-on-the-Lake, Canada*, pages 369–374, 2002.