Separating Handwritten Material from Machine Printed Text Using Hidden Markov Models

Jinhong K. Guo and Matthew Y. Ma
Panasonic Information and Networking Technologies Laboratory
2 Research Way, Princeton, NJ 08540, USA
(kguo, mma)@research.panasonic.com

Abstract

In this paper, we address the problem of separating handwritten annotations from machine printed text within a document. We present an algorithm that is based on the theory of hidden Markov models (HMM) to distinguish between machine printed and handwritten materials. No OCR results are required prior to or during the process and classification is performed on a word level. Handwritten annotations are not limited to marginal areas as the approach can deal with document images having handwritten annotations overlaying on machine printed text and shown to be promising in our experiments. Experimental results show that the proposed method can achieve 72.19% *recall for fully extracted handwritten words and* 90.37% *for partially extracted. The precision of extracting handwritten words reaches* 92.86%.

1. Introduction

A document image may contain a mixture of machine printed text and non-machine printed text such as handwritten annotations. Machine printed text reflect the originality of the document while the add-on contents (such as handwritten annotations, stamps etc.) reflect the alternation that has been done to the original document. Once added-on information and original content of a document are identified, various applications can benefit. For example, the two kinds of information within the same document can be passed to appropriate recognition engines such as OCR for machine printed text and ICR for handwritten annotations, for efficient recognition. The separation of handwritten annotations may also benefit form processing or bank check processing in which only handwriting is of interest. In this paper, we present a novel approach based on Hidden Markov Model (HMM). In Section 2, we describe some related work and outline our approach. We introduce our hidden Markov model and discuss our approach in Sections 3 and 4. Experiments and conclusions are discussed in Section 5 and Section 6, respectively.

2. Related work and our approach

2.1. Literature review

Several approaches focused on improving OCR accuracy by detecting machine printed text and handwritten text for feeding into separate recognition engines [6][10][11]. However, they have certain limitations. Umeda *et al.* [10] make assumptions on the uniform block of either handwritten or machine printed text. They use pre-defined template to calculate vertical, horizontal and slant counts in order to determine whether a block belongs to machine printed or handwritten text. Violante *et al.* [11] make similar assumptions on the uniform block and use similar templates to calculate vertical and horizontal features of a block. A neural network classifier is then used. Because of similar horizontal and vertical features that have been used, both Umeda and Violante's work tend to be sensitive to the variation of machine printed text such as fonts, style and size. The algorithms can not be readily applied to a document with mixed machine printed text and handwritten annotations. Pal *et al.* [6] developed a method to separate handwritten text from machine printed text for Bangla and Devnagari and used specific uniform line features of these particular languages.

In form processing, attempts with the aid of marker/template appeared in some specific applications. For example, Graf *et al.* [2] used a method for compressing images of financial instruments such as bank checks by separating handwritten text from the static check form. This method is entirely depending on a "document identifier", from which other fields that text might appear can be located. In case of bank check, the "document identifier" is the MICR line - a magnetic ink character recognition line at the bottom of the check. The fact that

0-7695-1263-1/01/$10.00 © 2001 IEEE

a "document identifier" is required limits the application to very special fields such as bank checks, x-ray and NMR images. In reality, a document management system may have to handle various types of documents: business letters or forms, images, fax documents etc.

In our previous work [5], a novel method based on projection profile and line merge were proposed. This method, however, can only handle handwritten annotations located beyond document margins or between paragraphs but have limitations on handling handwritten text that can overlay anywhere in a page.

2.2. Our approach

Our approach to dealing with handwritten material separation is to consider statistical variations of appropriate features. We hypothesize that in machine printed text we will observe a large number of regularities on the projection profile (to be defined later) because of regularities in machine printed text. On the other hand, handwritten annotations tend to vary by style, author and environment such that they appear more irregularly. This can be shown in Figure 1. In classification, we perform the discrimination on a word level. It is not necessary to descend to the character level since a single word (or string) is typically uniform with respect to style. Our algorithm is based on the theory of Hidden Markov Models (HMM). The unknown OCR knowledge is treated as the hidden states and the decision is based on the observation sequences that come from the states. Details of our new approach are described in Sections 3 and 4.

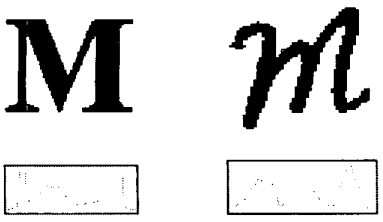

Figure 1. Illustration of projection profile for machine printed and handwritten text.

3. The HMM for handwritten material discrimination

3.1. Our model

The rich mathematical structure of HMM have led to their use in many applications. One of the most successful applications is speech recognition [4][8], but the idea has also been applied to optical character recognition [7][9] and keyword identification [1].

In natural language, there is an embodied Markovian structure. For example, the probability of seeing a letter "u" after letter "q" is usually greater than seeing "u" after any of the other letters. In a discrete Markov process, each state corresponds to an observable deterministic event. But in a hidden Markov model, the output of each state corresponds to an output probability distribution. In a hidden Markov model, the observation is a probabilistic function of the state. The HMM is a doubly embedded stochastic process with an underlying stochastic process that is hidden, namely the identity of the character in our scenario.

Let N be the number of states in the model and M the number of distinct observation symbols per state. A is the state transition probability matrix, where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i],\ 1 \leq i, j \leq N$$

B is the observation symbol probability matrix, where

$$b_j(k) = P[v_k \ at \ t \ | q_t = S_j],\ 1 \leq j \leq N,\ 1 \leq k \leq M$$

π is the initial state distribution, i.e.

$$\pi_i = P[q_i = S_i],\ 1 \leq i \leq N$$

Given the observation sequence $O = O_1 O_2 \ldots O_T$ and the Markov model $\lambda = A, B, \pi$, we wish to compute the probability $P(O|\lambda)$. In order to do this, a classical Forward-Backward procedure can be used.

In our model, we use 62 hidden states corresponding to the 52 upper and lower case Latin characters and 10 numerical digits. We use 10 symbols corresponding to 10 levels of projection values in observation sequences. Thus, the states are the unknown letters and the observation sequence is a probabilistic function of the states. In this model, element a_{ij} of the state transition probability matrix A is the transition probability between two neighboring letters within a word; element b_j of B is the probability of a symbol occurring in a given letter or digit, and element π_i of π is the probability that the word starts with that given letter or digit.

3.2. Training

The purpose of training is to obtain the HMM model for machine printed text (λ_1) and handwritten text (λ_2). In training, the probability matrices A and π are obtained from samples of ASCII text. The matrix B is computed from a set of bitmap images of various styles of letters and digits. The matrices A and π are the same for both machine and handwriting models while matrix B has to be computed separately.

In training matrix A, we used a sample of English text consisting of 5 pages. We computed the statistics of transition probability between two neighboring letters within a word. We also used the same sample to compute the probability of each state (letter or digit) starting a word, to construct matrix π.

Figure 2. Training sample of handwritten characters.

In training matrix B, samples of individual letters were obtained for machine printed and handwritten text, respectively. An example of handwritten training sample is shown in Figure 2. Multiple samples were collected for each individual letter. For every sample of each letter, a projection profile is calculated and the probability is computed on an averaging basis.

The projection profile for a letter is defined as a vertical projection of image pixels within its bounding box. An example of projection profile is shown in Figure 1. Assuming the number of black pixels at a position j of a letter is $S(j)$, the probability of this specific profile is

$$P_j = \frac{S(j)}{\sum_{k=0}^{N-1} S(k)}, 0 \leq j \leq N-1$$

where N is the width of the letter in pixel. The probability value P_j is quantized into M levels, $M = 10$ is the number of symbols in our model. Accordingly, for each letter, the probabilities of each symbol can be computed.

In training, 27 samples were collected for each machine printed letter (state) in order to take into account the various font styles. For handwritten letters, 9 samples from several different users were collected for each handwritten letter, to consider various ways people write. In addition, the method of computing symbol probabilities is font size independent. In our training, all samples for computing matrix B are scanned at $300dpi$.

4. Our approach

4.1. Segmentation of words

Our approach relies on the identification and segmentation of text strings in the image in order for HMM to work on the word level. A number of algorithms for segmentation of text blocks appear in the literature[3]. In our approach, we first generate connected components. Then based on the regularity on height and distance between neighboring letters, we merged these connected components into words. In segmentation, small components (such as a dash) are eliminated before grouping.

4.2. Observation sequences

At word level, a segmented word may consist of several individual letters that are represented by bounding boxes. For each letter within a word, a projection profile is computed and quantized into M (number of symbols) levels. An observation consists of a sequence of these M-level values. At word level, the observation sequence is obtained by concatenating projection profiles from individual letters that are within the word. The observation sequence of a word can be expressed as:

$$O = Q_1(1)Q_1(2)...Q_1(N_1)...Q_L(1)Q_L(2)...Q_L(N_L)$$

where L is the number of letters in a word, N_i is the width of letter i in pixel. $Q_i(j)$ is of value $[0, M)$.

4.3. Classification

The block diagram of classifying machine and hand generated text is shown in Figure 3. A document page is first grouped into text strings (words) as discussed in section 4.1. Then, projections of each letter within a word are computed and concatenated to generate the observation sequence of such word. The probabilities of observing the sequence given the models ($P(O|\lambda)$) are computed for each word given a particular model, currently machine printed text or handwritten text. The model that yields the highest probability is selected as the recognized class. As can be seen, two models are generated for our classification purpose: machine model or handwriting model. Once all words are classified, a post-processing procedure is utilized in order to improve the accuracy.

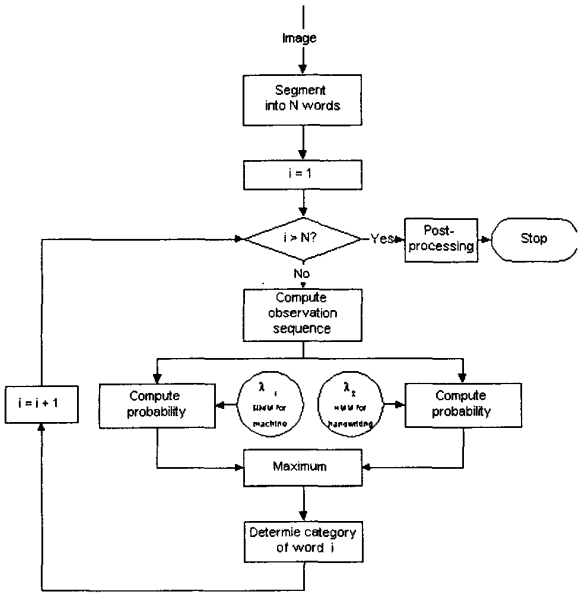

Figure 3. Block diagram for discriminating between machine and hand generated text.

4.4. Post-processing

Once all words are classified independently from HMMs, there maybe some errors due to various reasons, such as the error from word segmentation as well as the limited size of training data for HMM. In post-processing, single errors maybe corrected by examing the properties of their neighboring words. For example, if a word classified as handwritten text is nested nicely in a machine printed text line and its neighbors are machine printed, this word will be relabeled as machine printed. On the other hand, if a word classified as machine printed has no line forming information and is surrounded by handwritten words, it will be re-classified as handwritten text.

5. Experimental results

We have collected 25 document samples with total 187 handwritten words. These documents contain various font types and sizes. For each image, we run our HMM procedure and extract only handwritten text. Because handwritten words maybe written slanted, and letters within a word are much different in size, handwritten text may not be segmented well using the line merge technique. This will cause a handwritten word to be consisting of several parts, each being recognized either as handwritten text or machine printed text. If all parts within a handwritten word are recognized as handwritten text, we call it fully extracted. If some parts within a handwritten word are recognized as handwritten text, but not all, we call it partially extracted.

In our experiment, among 187 handwritten words, 135 words are completely extracted using the proposed method while 18 are completely missed. The recall is 72.19% for fully extracted handwritten words, and 90.37% for partially extracted handwritten words. Among 182 words that are recognized as handwritten text, 169 words are truely handwritten words. This yields a precision of 92.86%.

Figure 4 shows an original document image that has both machine printed and handwritten text. The resulting extracted handwritten text is shown in Figure 5 in bounding boxes. In the example, machine printed texts with various font styles and sizes were correctly classified, and handwritten annotations appearing at various locations within the document was correctly located. Handwritten annotation overlaid on machine printed text can still be identified. The images in Figure 4 and Figure 5 were cropped for viewing purpose in this paper.

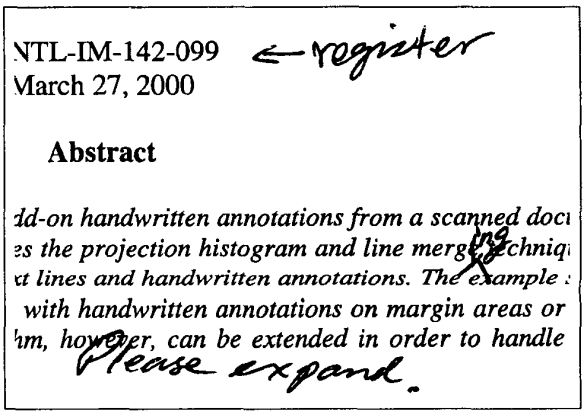

Figure 4. Original document with handwritten annotations on a machine printed page.

All documents were scanned at 300*dpi* or 400*dpi* using Panasonic KV-S6045 high speed scanner. The handwritten annotations were made using medium ball pen. The program was running on Panasonic CF-L1 laptop (Japanese model) with Pentium III 500MHz processor. The processing time without optimization is approximately 10 seconds per page.

6. Discussion and conclusions

In this paper, we have proposed a machine printed and handwritten annotation discrimination algorithm based on Hidden Markov Model (HMM) theory. The classification

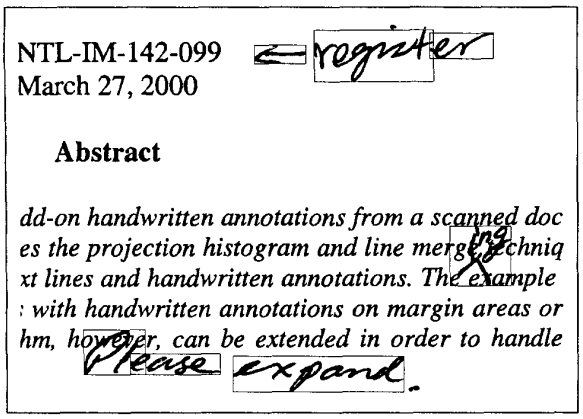

Figure 5. Extracted handwritten text labeled with bounding boxes.

was performed on a word-by-word basis thus it is robust and efficient in terms of processing time. This method does not rely on global information of a page. The results were obtained using very small set of training data. The proposed approach is shown promising. The machine printed text can vary in font style and size and handwriting can appear anywhere on a document, including overlapping on printed text.

The current algorithm still has certain limitations. Due to the nature of the HMM defined in the approach, it only applies to English or other Latin languages with training. It will be difficult to fit into the structure of such languages like Chinese and Japanese. One of the future research is defining a more general model that can be extended to other languages as well.

Additionally, we did not consider situations in which graphic information presents. More research can be done in the future to explore documents with mixture of text, graphics and handwritings. Future research may also include the study of features and observation sequence as well as training process and post processing etc.

For handwritten overlaid on machine printed text, once the handwritten annotation is identified, the removing (cleaning) of handwritten from machine printed text can be very useful yet a challenge task. Finally, the reliability of HMM may further be improved by re-estimation via a feedback training process.

References

[1] F. Chen, L. Wilcox, and D. Bloomberg. Detecting and locating partially specified keywords in scanned images using hidden Markov models. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 133–138, 1993.

[2] H. Graf and D. Mayer. Method and apparatus for separating static and dynamic portions of document images. U.S. Patent 5,631,984.

[3] F. Hoenes and J. Lichter. Text string extraction within mixed-mode documents. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 655 – 659, 1993.

[4] X. Huang, Y. Arki, and M. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh Unversity Press, 1990.

[5] M. Ma and K. Guo. Detecting and utilizing add-on information from a scanned document image. Technical Report PINTL-IM-142-099, Panasonic Information and Networking Technologies Labortory, 2000.

[6] U. Pal and B. Chaudhuri. Automatic separation of machine-printed and handwritten text lines. In *Proceedings of 5th ICDAR*, pages 645–648, 1999.

[7] H. S. Park and S. W. Lee. On-line recognition of large-set handwritten Hangul with hidden Markov models. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pages 51– 61, 1993.

[8] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pages 257–286, 1992.

[9] B. Sin and J. Kim. A statistical approach with HMMs for on-line cursive Hangul (Korean script) recognition. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 147 – 150, 1993.

[10] T. Umeda and S. Kasuya. Discriminator between handwritten and machine-printed characters. U.S. Patent 4,910,787.

[11] S. Violante, R. Smith, and M. Reiss. A computationally efficient technique for discriminating between handwritten and printed text. In *IEEE Colloquium on Document Image Processing and Multimedia Environments*, pages 17/1 – 17/7, 1995.