RECOGNITION OF CURSIVE WRITING ON PERSONAL CHECKS

ANDRÁS KORNAI, K.M. MOHIUDDIN
IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120

SCOTT D. CONNELL
Dept. of Computer Science, Michigan State U.,East Lansing, MI 48824

The system described in this paper applies Hidden Markov technology both to the task of recognizing the cursive *legal amount* on personal checks and the isolated (numeric) *courtesy amount*. Throughout the paper, our primary goal is to present methods that will allow the engineer to gain maximum leverage from a limited amount of training data.

1 Introduction

In this paper we describe the main components of a bank check amount recognition system prototype developed at the IBM Almaden Research Center. The goal of the system is to enhance the performance of a commercially available check recognition system by performing recognition not only on the isolated (numeric) *courtesy amount* but also on the cursive handwritten *legal amount* that appears on standard US personal checks. In addition to the core handwriting recognition task, such a system presents an additional challenge, the reliable segmentation of the image into recognizable units.

Section 1 presents the principal preprocessing steps including deskewing, zone finding, and feature extraction. Section 2 describes the combined segmentation and recognition module, which is a finite state grammar driven Hidden Markov Model (HMM) performing Viterbi segmentation and recognition, and introduces a simple measure of segmentation correctness that is a good predictor of overall recognition correctness. Results are presented in Section 3.

2 Preprocessing

The input bilevel TIFF images having the dimensions of the standard US personal check (6 by 2.7 inches, 1440 by 648 pixels at 240 dpi) are captured by a high speed/high volume commercial check processor. As in many other domains, e.g. in postal and in forms processing applications, once the infrastructure for tapping into a live data stream is established, raw images are no longer a scarce resource. The most serious bottleneck for statistical methods in such cases is the lack of *ground truth* data.

For example, fully automatic training of a skew estimator would require an independent source of skew angle: either a perfect automatic deskewer that is already available (in which case the point of training a new one is moot) or, more realistically, a human using a protractor. Since manual estimation of skew angle is rather labor-intensive, we could only gather a small *seed set* of skew-truthed images. While the skew estimator that we debugged and refined starting with this seed set uses fairly standard methods (a variant of the Postl method[1] enhanced by calculations based on beginning- and endpoints of horizontal black pixel runs), the bootstrap methodology itself is worth describing, as it was used throughout the development of the system.

> Stage 0 is the manual creation of a seed set, and the development of a working prototype. In stage i, a larger *current set* is generated using the stage $i-1$ module and verifying its output both by manual spot-checking and by passing it to subsequent modules. Systematic errors are corrected in the current set, and the stage i module is trained/debugged using this set.

Errors in skew detection were often detectable as statistical anomalies (outliers) after feature extraction – in a system of pipelined modules, errors of early modules that do not show up as statistical anomalies later in the pipeline tend to be irrelevant.

The second module of our pipeline is the zone finder, which deploys the same algorithm[2] with different settings for the legal and the courtesy zones. The legal zone is found chiefly on the basis of the preprinted legal and pay_to lines on the check, which are generally detectable as peaks in the row projections along the skew angle. The courtesy zone can be detected from the valleys (lack of black pixels) in the row projections. Starting from a seed set of a few dozen images, the stage 1 zone finder was developed over several hundred, and the stage 2 zone finder over several thousand images.

The third module, feature extraction, simulates the temporal progression found naturally in speech and dynamic handwriting data by spatial progression along the x axis. A sliding window of height h, width w, and slant k is used to sample the image with stepsize s. In most experiments, the height of the zone is normalized to 24, 12, or 6, by means of subsampling. In linear subsampling, every c horizontal lines are replaced by a single horizontal line which has a black pixel wherever any of the original lines had black. In nonlinear subsampling, c increases with the distance from the regions of greatest interest, so that descenders and ascenders get squeezed into fewer lines than strokes within the central region. In either case, blackness-preserving subsampling can be replaced by computing an average gray value. The result is

a vector of h/c dimensions providing a crude grayscale image of the original window. Currently width is 16 or 8, and the stepsize is 8 or 4, so that every point appears in exactly two halfway overlapping windows. To further reduce the dimensionality for the HMM stages (as well as for other algorithms), principal component analysis was performed. Using the IBM Hawthorne on-line recognition system [3] we projected the feature vectors onto the space spanned by the eigenvectors corresponding to the d largest eigenvalues of the overall covariance matrix. In most experiments, d is between 8 and 32.

3 Segmentation and recognition

In traditional OCR systems segmentation and recognition are performed by two separate modules. The segmenter analyzes the image using connected components and other heuristics, and passes the coordinates of the bounding boxes to the recognizer. Typically, several segmentation alternatives are proposed by the segmenter, and the recognizer acts in part as a postprocessor that rejects segmentation hypotheses that contain unrecognizable (or recognizably wrong) image segments. In HMM-based systems the segmentation and the recognition are performed in parallel, in a Viterbi search that takes into account not only the segmentation alternatives and their recognition scores, but also the combinatorial restrictions characteristic of the domain sublanguage. Of the three systems A,B,C voted here, system C is a traditional handprint digit recognizer enhanced by special symbols (such as the \$ sign and the "c" (century) sign for two tightly written zeros) and heuristics specific to the courtesy amount domain. System A is the HMM legal amount recognizer [4], and system B is a courtesy amount recognizer also employing HMM technology.

In our experience, the overall success rate of the system can be expressed as a product of two factors: the success rate of the segmenter and the success rate of the recognizer. More formally, given an unsegmented image I such as the courtesy or the legal field, let us denote the portion falling between x coordinates s and s' by $I(s, s')$. Assume correct segmentation $s_0, s_1, ..., s_n$, and correct labels $l_i = L(I(s_{i-1}, s_i))$ for $i = 1, ..., n$. If the segmenter returns the segmentation $t_0, t_1, ..., t_m$ with probability $P(t_0, t_1, ..., t_m|I)$ and the recognizer returns the label r with probability $Q(r|I(s, s'))$, the probability of correct recognition is

$$\sum_{m=1}^{\infty} \int_{t_0,...,t_m} \cdots \int P(t_0, ..., t_m|I) \prod_{i=1}^{m} Q(l_i|I(t_{i-1}, t_i))dt_i$$

m denotes the number of segments returned by the segmenter and the integral

is taken over all segmentation hypotheses with m segments. If the recognizer performs no segmentation (i.e. it never returns a recognition hypothesis with zero or with more than one segment in the output) and the effects of post-processing are not counted here, the only term in the series that will actually contribute to the probability of correct recognition is the one for $n = m$. In that term, the probabilities $Q(l_i|I(t_{i-1}, t_i)$ fall off rapidly if t_{i-1} and t_i are far from s_{i-1} and s_i respectively, so the only volume of n-space that we need to consider in evaluating the integral is a small n-cube given by points $t_0, t_1, ..., t_n$ satisfying $|t_i - s_i| \leq R$ for all $0 \leq i \leq n$. The maximum r of $|t_i - s_i|$ is therefore a good measure of error for any given segmentation $t_0, t_1, ..., t_m$ relative to the the ground truth $s_0, s_1, ..., s_n$.

It is well known that heuristic algorithms (e.g. searching for peaks and valleys) for character level segmentation of cursive writing are not reliable enough for statistical pattern matching. What is perhaps more surprising is that segmenting cursive writing *at the word level* already poses serious difficulties for such heuristic segmenters. To get a better sense of the overall difficulty of the problem, we computed r for a large test set, and plotted what percentage of the images that have r below a tolerance of 4,8,... pixels.

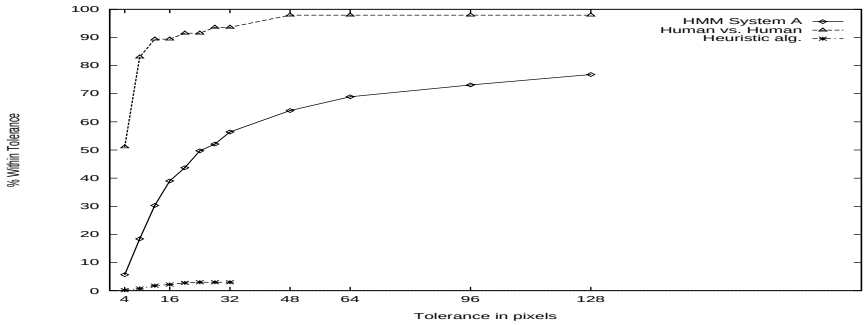

Figure 1: Segmentation error as a function of tolerance

As Fig. 1 shows, human segmenters compared to "ground truth" provided by other human segmenters will get about 50% of the images correct for very narrow tolerance ($r \leq 4$), but their results are consistent with one another for over 90% of the cases if we tolerate 16 pixels error (1.7 mm). Segmenters based on the usual heuristics employing connected components and vertical projection profiles will get no more than 4% of the images correct within $r \leq 32$ pixels (3.4 mm, comparable to average character width in our sample). This number is so low because the task of finding the left edge of the first content word and the right edge of the last content word requires a rudimentary ability

to distinguish *content words* such as "fifteen" from *function words* like "and" or "cents". The ability of performing segmentation and recognition in parallel is what distinguishes HMM systems such as System A, which is over 55% correct within $r \leq 32$ pixels, from more traditional heuristics-based systems.

Legal field images contain a great deal of noise. In addition to the hand-written content and function words, checks will contain non-numeric written material, such as "&" or the horizontal line used to fill spaces, preprinted material, such as the word DOLLARS and portions of the text PAY TO THE ORDER OF hanging down from the line above. They also contain edge noise, partly preserved from the preprinted box that surrounds the body of the check, and partly created by the scanning process itself. In the grammar describing the sublanguage of the check domain, the start symbol is rewritten as *Enoise Lnoise Body Lnoise* (FR) *Lnoise* DOLLARS *Lnoise Enoise* – only the *Body* refers to material with actual numerical content that we wish to recognize.

To bootstrap word-level truthing, first a seed set of two thousand images were manually segmented from beginning to end, and an HMM recognizer was trained. In stage 1, we used the stage 0 HMM to segment the images and select only the body. This way, the average length of the image to be hand-segmented was reduced by two-thirds, and valuable human resources no longer had to spent on hand-segmenting irrelevant material. The stage 1 HMM, trained on several thousand images, was applied in a similar fashion in stage 2, when slant-estimation and correction was first applied. In earlier stages, estimation of the slant would have been problematic, because only the content words present a consistent slant profile.

4 Results

Figure 2 shows the error rate (plotted on the y axis) as a function of reject rate (plotted on the x axis) for system C in standalone mode and in combination with A or B. The raw character error rate of system C is over 9.7%, so the whole courtesy field, which has on the average 4.12 content characters (plus the decimal point) is not expected to be correct in more than 60% of the cases. By aggressively rejecting 76% of the input checks, those where the recognizer produces less than full confidence output, system C can be made to perform at a 2.56% field error rate. By voting with system B, which has a 25.9% raw character error rate, the same 2.56% field error can be accomplished at a 59% reject rate i.e. we can nearly double the number of accepts.

As can be seen from Figure 2, the error-reject curve for A+C lies significantly below the C curve. At 50% rejection, system C had 4.18% error, while the combined system has 2.75%, less than two-thirds of the original error. To

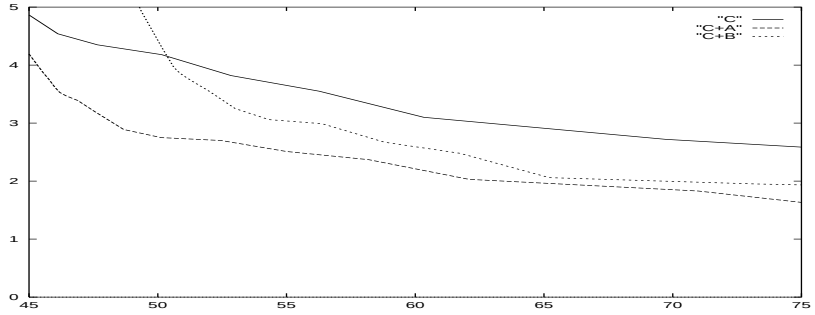

Figure 2: Error rate as a function of rejection rate for systems C, C+A, and C+B

replicate this error rate, system C would need to reject not 50% but 75% of the data. The error-reject curve for B+C shows improvement over C only in a narrower range, and the improvement is not as marked as for A+C. The reason for this is that B and C use the same courtesy image, while A and C operate on different images and therefore exploit a larger information base.

Acknowledgments

We would like to acknowledge the positive impact of many discussions held with the on-line handwriting recognition group at IBM Hawthorne, in particular with Krishna Nathan, Jayashree Subrahmonia, and Homayoon Beigi. We benefited a great deal from their advice and from our study of their system.

References

1. W. Postl "Detection of linear oblique structures and skew scan in digitized documents" *Proc. 1986 ICDAR* 687-689
2. A. Kornai and S.D. Connell, "A zone finding algorithm for checks" *Proc. 13th ICPR* 1996, Vol III, 818-822.
3. J.R. Bellegarda, D. Nahamoo, K.S. Nathan and E.J. Bellegarda, "Supervised Hidden Markov Modeling for On-line Handwriting Recognition," *Proc. 1994 ICASSP,* Adelaide, South Australia, Vol 5, pp 149-152, 1994
4. A. Kornai, K.M. Mohiuddin and S.D. Connell, "An HMM-based legal amount field OCR system for checks," *Proc. Systems, Man, and Cybernetics* 1995, 2800-2805.