AN EXPERIMENTAL HMM-BASED POSTAL OCR SYSTEM

András Kornai

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120
kornai@almaden.ibm.com

ABSTRACT

It is almost universally accepted in speech recognition that phone- or word-level segmentation prior to recognition is neither feasible nor desirable, and in the dynamic (pen-based) handwriting recognition domain the success of segmentation-free techniques points to the same conclusion. But in image-based handwriting recognition, this conclusion is far from being firmly established, and the results presented in this paper show that systems employing character-level presegmentation can be more effective, even within the same HMM paradigm, than systems relying on sliding window feature extraction. We describe two variants of a Hidden Markov system recognizing handwritten addresses on US mail, one with presegmentation and one without, and report results on the CEDAR data set.

1. INTRODUCTION

Any approach to speech and handwriting recognition must take into account that the signal is composed from a succession of alphabetic units (phonemes or graphemes). In the early work on speech recognition [13] as well as in character recognition this understanding led to systems that divided the overall recognition task into three separate tasks: segmentation, recognition, and postprocessing (context checking), performed in a cascade fashion. The realization that prior to recognition it is virtually impossible to segment the signal into phone-size units led to the broad acceptance of Hidden Markov techniques in the speech domain, since HMMs effectively delay segmentation decisions until the context checking stage [1].

In the OCR domain the issue is more complex. Historically, commercial products were first deployed for machine print and typewritten material, where segmentation is considerably easier than in speech. Only in the last few years has the reliability of isolated letter recognition reached the level where errors of segmentation became the dominant source of error [3]. In the domain of postal address recognition, the high volume (millions of mail pieces per day) makes possible very high savings in mail handling costs even for *selective* systems which deal only with machine-print and typewritten addresses and reject the rest, be they cursive, touching, or just low quality isolated handprint. Such selective systems, typically rejecting over a third of their input, have been deployed by large postal service throughout the world.

In this paper we will ignore earlier stages of processing up to and including the modules that find the address block and separate the lines [8], noting only that (1) these steps can be performed on a drastically subsampled image, containing no more than 1 pixel for every 8 by 8 region of the original image given at 200-240 ppi resolution and that (2) establishing the baseline of writing, including line-by-line skew, is more important than establishing the outer (ascender and descender) lines, since the outer regions are deemphasized during feature extraction. In conventional systems, the remaining stages of word and character-level segmentation, recognition, and postprocessing are disjoint modules arranged in a cascade, which makes late rejection and sending the data upstream for re-evaluation by earlier modules a major architectural challenge.

Our earlier experiments on isolated character data [9] demonstrated that given perfect segmentation Artificial Neural Nets (ANNs) are more efficient recognition devices than HMMs. On the other hand, ANNs are very inefficient as segmentation devices. Therefore, in systems dealing with less than perfectly isolated handwriting, the techniques at the heart of the HMM paradigm, in particular dynamic programming, are of great relevance. In the image-based OCR domain we are faced with a design tradeoff between the more efficient (implicit) segmentation and the less efficient recognition provided by HMMs.

To bring the issue into sharper focus, we created a framework in which the presence or absence of a dedicated presegmentation component corresponds to two different methods of feature extraction, but otherwise the two systems are the same. In Section 2 we describe the segmentation-free (sliding window) feature extraction method used in our experiments and elsewhere [10], and in Section 3 we describe another, connected component based method. The results obtained by the two are compared in Section 4 and conclusions are offered in Section 5.

2. FEATURE EXTRACTION BY THE SLIDING WINDOW METHOD

Here and in what follows we assume that recognition of a full line is to be performed: in the postal domain such a line can contain a name, a street address or a city, state, and zip. For the sliding window method, the line image is first height-normalized to 64 pixels. Since the standard algorithms for scaling a bilevel image yield greyscale output, some form of binarization becomes necessary even for those images that start out as binary on the CEDAR CD-ROM [6]. We investigated two scaling algorithms, bilinear and bicubic resampling. Linear resampling was faster, but only marginally, and the results were indistinguishable as far as later stages and overall recognition accuracy are concerned. We also investigated two different binarization methods, global thresholding and local (adaptive) thresholding [11]. Here the impact was marked: local thresholding was considerably better than global. To obtain comparable results for the greyscale CEDAR data, we added a binarization stage prior to scaling, and investigated all eight combinations: first local vs. global thresholding, next bilinear vs. bicubic resampling, and finally again local vs. global thresholding. To summarize our results, the less global thresholding done the better.

The scaled (re)binarized image is noise-cleaned: in particular underscores and dashed underscores are removed. Manual inspection reveals that this process leaves no visible traces of removal in about 97% of the cases. Next, the top and bottom of the writing is computed for every eight pixel wide column, and the results are low-pass filtered over five columns encompassing 40 vertical pixels. This results in images where ascenders or descenders are never cut off, but if they are not present, the 'n' zone, i.e. the body of the ascender- and descender-less letters such as *a c e i m n o r s u v w x z,* fills the whole image. The effect of getting more of the n-zone into the feature vectors is enhanced by the next stage of nonlinear resampling, whereby a column of 64 vertical pixels is reduced to a set of 8 greyscale values, but the values close to the center are based on fewer pixels than far from the center. In effect, we condense the image by a factor of 12 at the edges, but only by a factor of 4 in the center.

By this process, the image strip has been replaced by a sequence of 8-dimensional real vectors whose progression in time corresponds to progression in space along the x axis of the original line image. In the final stage of feature extraction this sequence $v_0, v_1, \ldots, v_k$ is first triplicated, meaning that each v_i is replaced by a concatenation v_{i-1}, v_i, v_{i+1}. This yields a sequence of 24-dimensional vectors which are reduced by principal component analysis to 12-16 dimensions [2]. The main additions to our earlier feature extraction method which was successful in a bank check recognition system [10] are the initial global height normalization and the lowpass filtering of local height.

3. FEATURE EXTRACTION BY PRESEGMENTATION

In the zipcode field, numbers are rarely touching, and one field, such as state, rarely touches another, such as city or zip. These observations lead to a system where the primary unit of analysis is the *connected component,* and the expectation that word boundaries will also be connected component boundaries. Though this expectation is largely met, both cursive writing and touching handprint require that connected components be cut into smaller parts that we will call *frags*. In the segmentation algorithm developed by Jianchang Mao and Prasun Sinha at IBM Almaden, and used in the experiments, the central heuristic used for the cutting is the location of valleys (local minima) in the contour. A feature vector is computed for each frag using the contour direction features described in [12].

Given that the goal of the algorithm is to presegment characters, it is not surprising that for the majority of characters it creates a single frag, and therefore a single feature vector. The system oversegments, but only slightly: over two thirds of character tokens yield a single frag, 20% yield two, and less than .15% yields four or more. The average number of frags per character is 1.24, which makes single state character models a natural choice. This is in sharp contrast to the sliding window system, which must take into account that the width of characters varies widely, both within and across character classes, even after height normalization. To deal with across-class width variation, in the sliding window system models for different characters have different numbers of states ranging from 1 for dot (period) and 2 for i to 6 for w and 7 for m. For most characters we use Bakis models, with a self-loop for each state, a step transition to the next state, and a jump transition to the second state following it. If we enrich the model with a silent *input* state with transitions to any subsequent state [4], it becomes possible to fully absorb arbitrary width variation [7], and in certain classes, such as u or s, width variation was so extreme that we found it advantageous to do so.

To summarize the differences between the two feature extraction methods, in the sliding window system many (often more than a dozen) relatively low-dimensional feature vectors are extracted for each character, while in the presegmentation-based system only a few (typically only one) vector will be extracted. Since this vector is much bigger (originally 88 dimensions, in most experiments reduced to 48 or 32 dimensions by principal component analysis), the overall bitrate of the two feature extraction front ends is quite similar, about 4-8 bytes per horizontal pixel. This bitrate, being an order of magnitude larger than that required for pen- or tablet-based recognition, offers a rough measure of the difficulty of extracting dynamic information from an image.

4. TRAINING AND TESTING

The CEDAR CD-ROM [6] is organized with the requirements of the classical presegmentation-based OCR systems in mind: there are plenty of isolated character training data, often isolated and truthed by hand, and most files embody presegmentation at least at the word level. As a result, it is less than fully ideal for training and testing HMM-based OCR systems, and it was not entirely possible to follow the training and testing protocol that is implied by the structure of directories on the CD-ROM. One particularly important limitation is that the size of the CEDAR data set does not allow for training context-dependent models, thereby depriving HMMs of a significant advantage.

Zipcodes provide a convenient testing domain where context-dependence plays a very limited role – as we have noted earlier, the vast majority of zipcodes is written without touching characters. Using the `goodbs` directory of the CD-ROM we initialized sliding window models with 32 mixture diagonal gaussians. When tested on the the training data these models are basically 100% good, implying that all the relevant aspects of the training data have been absorbed. Next, these models were retrained on the 5-digit or shorter files in `train/zipcodes` data, excluding the `bc` directory, which was only used for testing. The same process was repeated with flatly initialized models so as to make sure that no `bc` data in `goodbs` contaminates the results – this made no difference.

By their selection, all `bc` files are within a single zip code, 14222, so our method tested only 3 of the 10 digit models directly. However, the results, 55% field correct (using the trivial perplexity 10 grammar that embodies only the restriction that zipcodes have five digits) remained unchanged when more complex training and evaluation schemes, directly involving all 10 digits, were used. These results, though 5% better than the only comparable HMM results reported in the literature [5], are not competitive with ANN results on similar data. Therefore we proceeded on the assumption that zipcodes will actually be recognized by a traditional OCR system, and HMMs will only be used for city, street, state, and addressee recognition. Readers familiar with the CEDAR CD-ROM will know that it supports an evaluation method based on this assumption, namely the use of precomputed dictionaries.

For readers not familiar with this method the idea can be summarized as follows. The effect of an imperfect zip recognizer is simulated by taking the true zipcode, replacing some digit(s) by a wildcard, and checking in the USPS database of zipcodes which cities have zipcodes that fit the wildcarded pattern. Depending on the number and position of wildcarded digits, we can derive a set of language models of increasing perplexity. The CD-ROM provides three levels of difficulty: for the `bd` directory, which we used for all of the tests (and some of the training) the city lists provide an average of 14.3, 106.4, or 928.5 alternatives, and the state lists provide 2.4, 7.6, or 19.1 alternatives, for difficulty levels 1, 2, and 3 respectively. In evaluating different systems, we used these lists rather than generating our own, to maintain comparability with results reported elsewhere. Since some files had to be excluded because the true solution was not among the alternatives provided on the list, in the following table we report the average number of alternatives separately.

task	perplexity	reco rate
state1	2.9	84.5
state2	8.2	69.0
state3	19.3	53.9
city1	7.7	63.6
city2	50.3	33.8
city3	410.3	18.6

These figures could be significantly improved by allowing for a postprocessor that would resolve differences such as *BARRACKS,* vs. *BARRACKS* or *BEVERLY* vs. *BEV* that have no effect on the address. The recognition rates were obtained on the `train` part of the CEDAR CD-ROM, but we emphasize that the data used for training the systems came from an entirely different (non-CEDAR) source, which reflects a somewhat different (European) style of writing. Results are for the system that uses presegmentation – the system with sliding window features is on the average 10% worse.

5. CONCLUSION

How much does 10% difference in recognition rate support the overall conclusion that some kind of presegmentation is preferable to sliding window feature extraction? At the theoretical level, proponents of the sliding window can very well argue that the CEDAR CD-ROM does not constitute a level playing field, since it has relatively little cursive (as opposed to handprint), and does not have nearly enough data for training a sophisticated HMM system. At the practical level, however, our experiments leave little doubt that presegmentation, even though it requires significant computational effort compared to sliding windows, contributes greatly to the effectiveness of the overall system.

First, the zipcode experiment shows that in a domain where most of the characters are isolated, systems that do not take advantage of this fact are not competitive. In the bank check domain, where most of the characters in the legal amount field are connected, we found the opposite result: heuristic segmentation schemes based on connected components and vertical projections were considerably worse than sliding window systems [9]. Taken together, the zipcode and the bank check results imply that being isolated vs. being connected is a fundamental property of character image data, and OCR systems need to be tuned to this property just as they need to be tuned to other characteristics of the data domain.

Second, the city/state experiment shows that in the postal domain, as represented by the CEDAR CD-ROM, the handwriting found in the alpha fields is closer in its degree of isolatedness to numeric fields than to the cursive handwriting found in bank checks, which makes a presegmentation-based method the method of choice. To be sure, this leaves open the larger question of what is the appropriate method for purely cursive data, but until such data becomes the dominant portion of postal OCR rejects, or until the state of the art advances to the point that tuning to data characteristics is no longer necessary, practical considerations will continue to favor presegmentation.

6. ACKNOWLEDGMENTS

Comparing the two approaches would not have been possible without the support and advice of the other members of the Almaden OCR group: Sandeep Gopisetty, Raymond Lorie, Jianchang Mao, Moidin Mohiuddin, and Thomas Truong.

7. REFERENCES

[1] J.K. Baker, "Stochastic modeling for automatic speech understanding" Reprinted in A. Waibel and Kai-Fu Lee (eds) *Readings in Speech recognition*, Morgan Kaufmann, San Mateo CA, 1990, 297-307

[2] J.R. Bellegarda, D. Nahamoo, K.S. Nathan and E.J. Bellegarda, "Supervised Hidden Markov Modeling for On-line Handwriting Recognition" *IEEE Proc. ICASSP,* Adelaide 1994, Vol 5, 149-152

[3] M.R. Bokser, "State of the Art in a Commercial OCR System: A Retrospective View" Paper presented at the IS&T/SPIE Symposium on Electronic Imaging, San Jose CA 1996

[4] T.H. Crystal and A.S. House, "Segmental durations in connected speech signals: current results" *JASA* **83** 1988, 1553-1573

[5] A.J. Elms, *The representation and recognition of text using Hidden Markov Models.* University of Surrey PhD Thesis, 1996

[6] J.J. Hull, "Database for handwritten word recognition research" *IEEE PAMI* **16** 1994, 550-554

[7] A. Kornai *Formal Phonology.* Garland Publishing, New York, 1995

[8] A. Kornai and S.D. Connell, "Statistical Zone Finding" *IEEE Proc. 13th ICPR*, Vienna 1996, Vol III, 818-822

[9] A. Kornai, K.M. Mohiuddin and S.D. Connell, "An HMM-based legal amount field OCR system for checks" *IEEE Proc. SMC*, Vancouver, BC 1995 Vol 3, 2800-2805

[10] A. Kornai, K.M. Mohiuddin and S.D. Connell, "Recognition of cursive writing on personal checks" *Proc. 5th IWFHR*, Essex 1996, 373-378

[11] K.M. Mohiuddin and J. Mao, "A Comparative Study of Different Classifiers for Handprinted Character Recognition" *Pattern Recognition in Practice IV*, 1994, 437-448

[12] H. Takahashi, "A Neural Net OCR Using Geometrical and Zonal-pattern Features" *Proc. 1st ICDAR*, 1991, 821-828

[13] W.A. Woods *et al* "Speech understanding systems: final technical progress report" Bolt Beranek and Newman Inc. Report 3438, Cambridge MA, 1976