OCR with No Shape Training

Tin Kam Ho
Bell Labs, Lucent Technologies
700 Mountain Avenue, 2C425
Murray Hill, NJ 07974, USA
tkh@bell-labs.com

George Nagy
Dept of ECSE
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA
nagy@ecse.rpi.edu

Abstract

We present a document-specific OCR system and apply it to a corpus of faxed business letters. Unsupervised classification of the segmented character bitmaps on each page, using a "clump" metric, typically yields several hundred clusters with highly skewed populations. Letter identities are assigned to each cluster by maximizing matches with a lexicon of English words. We found that for 2/3 of the pages, we can identify almost 80% of the words included in the lexicon, without any shape training. Residual errors are caused by mis-segmentation including missed lines and punctuation. This research differs from earlier attempts to apply cipher decoding to OCR in (1) using real data (2) a more appropriate clustering algorithm, and (3) decoding a many-to-many instead of a one-to-one mapping between clusters and letters.

1. Introduction

In today's pixelated environment, any Tom, Dick and Jane can design or download the font that best conveys his or her message or personality. It is therefore of more than academic interest to liberate OCR from the stereotypic prototypes of predetermined character shapes. Document-specific OCR can learn the peculiarities of the dominant font much the way that we interpret a scrawled postcard by exploiting the similarity of letters or groups of letters in obscure words with those that appear in easily-read words.

Although there have been many earlier studies [1], [3], [4], [5], [8], [9], [10], [11], [19], [20], [21], [22] that exploited language context to decode character or word bitmaps, we believe that this is the first application of such techniques to a large collection of short, dirty documents. Our fax data, from the ISRI corpus, contains letterheads, addresses, signatures, upper and lower case, punctuation, underscores, and averages less than twenty lines of body type per document [25].

Neither the methods cited above, nor those developed expressly for substitution ciphers [6], [7], [13], [17], [18], [24], [26], [27], are robust enough to unscramble the many-to-many mappings encountered in the OCR application. Such mappings arise because impure clusters correspond to more than one alphabetic class of letters, and bitmaps corresponding to the same alphabetic class may appear in several clusters. We have developed a simple deciphering algorithm that is more effective for OCR than the classical methods.

Our work has benefited from renewed interest in symbol-based compression for the forthcoming JBIG2 standard [2], [12], [15]. Symbol-based text-image compression is typically twice as efficient as JBIG1 compression, which in turn is nearly twice as good as CCITT-G4. We therefore designed our OCR in the expectation of rapid adoption of the symbol-based text-image compression standards. Building OCR on top of symbol-based compression offers the benefit of dual-mode representation of documents [23] that allows search on the character-coded version and preserves the original page-image for viewing.

A critical advance in symbol-image compression has been the development of cluster-distance metrics that weight groups of adjacent difference pixels more heavily than an equal number of scattered difference pixels [16]. A further improvement that we introduce is the separation of difference clumps consisting of foreground pixels from clumps of background pixels. This metric, combined with a standard nearest-center clustering algorithm, improves significantly the purity of the resulting clusters.

The next section describes our data, preprocessing, clustering algorithm, decoding procedure, and evaluation. In the third section we present our results. In the conclusions we speculate on what is ahead in the direction that we have taken.

2. Data And Methodology

2.1. Data

The data consists of 200 English-language business letters transmitted locally in 204x196 dpi fine mode facsimile from a Xerox 7024 fax machine to a fax modem. The sample includes typewritten and poorly copied letters, some with handwritten annotations. The average number of words per letter is 257, and the average number of characters is 1600. 87% of the words could be found in our 21,466-word lexicon compiled from the Brown corpus.

According to [25], the average character accuracy of the tested commercial OCR systems barely topped 97%, in contrast with the nearly 99% obtained without facsimile transmission on the same documents scanned at 300 dpi. A character-level accuracy of 97% corresponds to a word accuracy of only about 85%.

2.2. Preprocessing

For layout analysis we follow [14]. We find the connected foreground components using 4-connectivity and merge some adjacent components like the dots on i's and j's. Text-line and word boundaries are determined using thresholds based on the average height of the connected components. At the end of this stage most of the character images are isolated, but some are conjoined and some are fragmented.

2.3. Unsupervised Classification

The first cluster is seeded by the first character-bitmap on the page. New clusters are created whenever a character bitmap cannot be assigned to one of the existing clusters. A character bitmap is assigned to the cluster to which its bitmap distance is least (distance is calculated only to the first bitmap of each cluster). The symmetric distance metric is computed by aligning two bitmaps to be compared according to their horizontal and vertical pixel medians, then shifting one relative to the other in a 3x3 neighborhood to find the best match. Because most of the bitmap pairs are highly dissimilar, the clustering algorithm has various bailout rules that allow it to abandon unpromising pairings quickly.

The symmetric distance between two aligned bitmaps A and B is defined as the asymmetric distance between A and B plus the asymmetric distance between B and A. The asymmetric distance between A and B is the count of the number of pixels that are black in A and white in B, with each black difference pixel in A weighted by the number of its black 4-neighbors in A. The asymmetric distance between B and A is the converse.

At the end of the initial clustering pass, similar clusters are merged if the ratio of their average intra-cluster to inter-cluster distance is larger than 0.5. The two averages are based on all pairs of bitmaps within the same cluster, and on all pairs in different clusters, respectively. The ratio is retained for further use.

Any singleton cluster is merged into the nearest larger cluster if its distance to any one of the members of that cluster is less than an arbitrary threshold value.

2.4. Context Analysis

Context analysis is done by iterative applications of several simple modules each attempting to assign labels to the clusters by different rules. Every tentative assignment is evaluated using a v/p ratio which is the number of valid words from the lexicon over the number of word-patterns containing them. A word-pattern may contain a mixture of elements from labeled and unlabeled clusters. Only one match will be counted even if there are multiple matches for the same word-pattern from the lexicon. Word interpretations are built up progressively from the accepted assignments. The most useful modules are as follows.

- JointAssign: We take the three largest clusters and try to assign to them every triplet of eight most common letters (observed in the Brown corpus) {a,e,i,o,n,r,s,t}, from which we select the triplet that maximizes the number of matching lexicon entries among all those words which contain at least two occurrences of these three clusters. For instance, by assigning e,i,o to clusters 1,2,3 respectively, the following strings (x standing for any other clusters) can be interpreted as the patterns below them and matched with words from lexicon in the third line:

```
clusters>  x3x13xx1x xxx3x2xx23x 1xx2x3
patterns>  _o_eo__e_ ___o_i__io_ e__i_o
matches >  homeowner association ??????
```

  The v/p ratio in this case is 2/3. If the best triplet makes a v/p ratio above 0.75, we accept the assignment. Otherwise, we try clusters 2,3,4, and 3,4,5 in turn.

- UniqueMatch: Next, every word-pattern containing at least one unlabeled cluster is checked to see if some assignment yields a unique lexicon match. For instance, the unlabeled cluster "_" in the pattern "w_ic_" will be tentatively labeled with "h" since it produces a unique match "which". The one in "_low" will not be assigned since the pattern matches both "flow" and "glow". The tentative assignment will be checked to make sure v/p is at least 0.25. The search is iterated with the updated patterns until no more new unique matches are found.

- MostMatch: If there are still unlabeled clusters, then the algorithm assigns every letter in turn to one of the unlabeled clusters and checks which assignments results in the highest v/p ratio. If the best ratio is at least 0.75, and the second best is not too close (at least 0.1 below), then this assignment is ratified. For example, if

cluster 9 appears in only four words, “9low,” “9ierce,” “a99air,” and “lu99a,” then f is assigned to 9 because it yields 3 lexicon words out of 4 (v/p = 0.75), whereas p results in only two matches (v/p = 0.5), and g in only one (v/p = 0.25) (if “luffa” appeared in the lexicon, then the match would be even safer).

- VerifyAssign: Finally, every assignment is verified by trying to replace it with each of the other 25 letters. If the v/p ratio can be improved, and either more than one word contains this cluster or the single word that contains it has at least two letters, then the label is replaced, and the verification is continued. Precaution is taken to guard clusters that appear only once as a single-letter word from receiving assignment of “a” or “i” unless context from other words would also justify it.

Other modules exploit the most frequent short words (1-4 letters) and the most frequent bigrams, or try to assign an unlabeled cluster to its nearest labeled neighbor determined by intra/inter-cluster distance ratio.

These modules make cumulative contributions in the interpretation process. Jointly, they are able to handle split clusters of the same symbol. Loose requirements (v/p less than 1) on the simultaneous assignment of all bitmaps in the same cluster give some tolerance for clustering errors that yield impure clusters.

2.5. Evaluation

The evaluation considers both the (partially) labeled sample and the ground-truth as a sequence of words without regard to line breaks. The two sequences are matched with the words as the basic units (two words have to be identical to be counted as a match) and the length of the longest common subsequence (LCS) is calculated (raw score 1). Then the interpretation is spell-corrected using the same lexicon for calculating another LCS score (raw score 2). To compare across different pages, we normalize raw scores 1 and 2 by the number of words in the truth file (called true words) to obtain final scores 1 and 2.

Business letters contain, of course, many proper nouns and digit sequences that can be identified only when their constituent letters share a cluster with bitmaps of lexical words. Errors in such assignments cannot be corrected by the spell checker. So we also count the true words appearing in the lexicon and normalize the raw scores with it to obtain final scores 3 and 4.

3. Results

The median proportion of characters per sample that are assigned alphabetic labels is 93% - the remainder are in clusters that cannot be matched to the lexicon. These include mis-segmented patterns, special symbols (e.g., $), digits, and punctuation. The median number of clusters per sample is 244; typically 55% of these are singletons.

Figure 1 shows a plot of the ratio of percent true words that are correctly interpreted (score 4). From the plot, we see that there are two clusters of results: most of the pages show up in a group with the scores above 50% and average near 80%, while the rest are below 50% and average near 20%. Table 1 lists the averages of all four scores broken down by these two groups. For pages in the first group, good knowledge of the letter content can be obtained from the word interpretations (see example in Figure 2). Pages in the second group suffered from catastrophic failures in the interpretation process so that no meaningful contents can be extracted. Recalling that shape based recognition of these pages achieved a word-level accuracy of only about 85% (before spell check), we believe that our method deserves further pursuit.

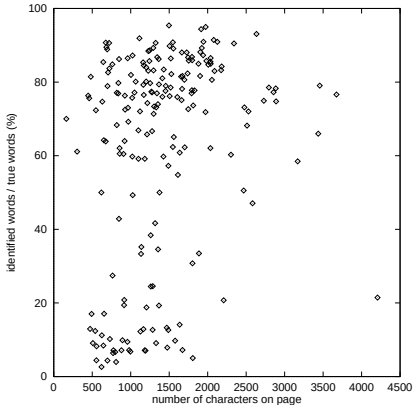

Figure 1. Percent true words (contained in our lexicon) identified versus number of characters on the page.

```
dear homeowners a an accordance with article add
of the bypass of the mess pillage homeowners
association a and paragraph a of the declaration
of a covenant a conditions a and restrictions for
the property a notice is hereby given that the
annual meeting of the mess pillage home owners
will be held at a a a a am on steady remember la
a saga at the recreation room located at ...
```

Figure 2. An example interpretation sequence after spell check.

4. Conclusions

We are convinced that adaptive, document-specific character recognition algorithms are necessary to improve OCR beyond its current plateau. Although commercial software performs very well on clean pages and on common fonts,

Table 1. Average scores by performance group. avg: average of scores 1 to 4.

group	#pages	score 1	score 2	score 3	score 4
avg ≥ 50	137	64.2	68.7	73.4	78.5
avg < 50	63	11.4	16.2	13.7	19.4
all	200	47.6	52.2	54.6	59.9

its error rate increases abruptly on low-quality pages and unusual typefaces that are easily read (in context) by humans. Context recognition based on the homogeneity of type shapes and image distortion within the same document is, of course, only one of the possible remedies. In this research, we explore how far linguistic context alone can take us. We expect that future systems will integrate contextual methods with shape based classifiers instead of restricting the use of context to post-processing.

The method that we have described can be readily applied to text images compressed with symbol-matching. Widespread acceptance of the standard will stimulate the development of special-purpose hardware. With methods such as those advocated here, the resulting volume of compressed text images can be efficiently converted to character-coded form without resorting to further pixel-level manipulation. Access to a standard file format for sequences of compressed character bitmaps will also greatly facilitate the development of OCR algorithms for specific applications.

The major weakness of our method is its inability to cope with digits, special symbols, and punctuation. Not only are these glyphs not recognized, but punctuation appended to a word precludes matching it correctly to the lexicon. Although it is clear that context is insufficient to recognize unconstrained, poorly digitized text, it is surprising that it comes fairly close to what has been achieved with shape-based methods.

However, contextual methods are also applicable to non-alphabetic symbols. We are currently attempting to extend contextual bitmap identification to this set. Since digits, special symbols and punctuation are seldom combined with letters or with each other in unique configurations, we expect that will have to rely more on statistical morphology than on strictly lexical methods. Fortunately, the availability of large corpora in coded form allows us to compile the necessary information.

Acknowledgements

Tin Ho thanks Jim Reeds for helpful discussions.

References

[1] R. Casey, G. Nagy, Autonomous reading machine, *IEEE Trans. Comput.*, **C-17**, 5, May 1968, 492-503.

[2] L. Bottou, P. Haffner, P.G. Howard, P. Simard, Y. Bengio, Y. LeCun, High quality document image compression with DjVu, *J. of Electronic Imaging*, **7**, 3, 1998, 410-425.

[3] R. Casey, G. Nagy, Advances in pattern recognition, *Scientifi c American*, 224, April 1971, 56-71.

[4] R. Casey, Text OCR by solving a cryptogram, *Proc. ICPR 8*, Paris, 1986, 349-351.

[5] C. Fang, J.J. Hull, A word-level deciphering algorithm for degraded document recognition, *Procs. SDAIR-5*, Las Vegas 1995, 191-202.

[6] A. Goshtasby, R.W. Ehrich, Contextual word recognition using probabilistic relaxation labeling, *Pattern Recognition*, **21**, 5, 1988, 455-462.

[7] G. Hart, To decode short cryptograms, *Commun. ACM*, **37**, 9, Sept. 1994, 102-108.

[8] T.K. Ho, J.J. Hull, S.N. Srihari, A word shape analysis approach to lexicon based word recognition, *Pattern Recognition Letters*, **13**, 1992, 821-826.

[9] T.K. Ho, Bootstrapping text recognition from stop words, *Procs. ICPR-14*, Brisbane 1998, 605-609.

[10] T.K. Ho, Fast identifi cation of stop words for font learning and keyword spotting, *Procs. ICDAR-5*, Bangalore 1999, 333-336.

[11] T. Hong, J.J. Hull, Improving OCR performance with word image equivalence, *Procs. SDAIR-5*, Las Vegas 1995, 177-190.

[12] P. Howard, F. Kossentini, B. Martins, S. Forchhammer, W.J. Rucklidge, The emerging JBIG2 Standard, *IEEE Trans. Circuits and Systems for Video Technology*, **9**, 7, November 1998, 838-848.

[13] D.G.N. Hunter and A.R. McKenzie, Experiments with relaxation algorithms breaking simple substitution ciphers, *Computer Journal*, **26**, 1, 1983, 68-71.

[14] D. Ittner, H.S. Baird, Language-free layout analysis, *Procs. ICDAR-2*, Tsukuba Science City 1993, 336-340.

[15] Joint Bilevel Document Image Experts Group, Progressive bi-level image compression, ITU recommendation T.82, ISO-IEC International Standard 11544, 1993.

[16] O. Johnsen, J. Segen, G.L. Cash, Coding of two-level pictures by pattern matching and substitution, *BSTJ*, **62**, 8, Oct. 1983, 2513-2545.

[17] S. Khoubyari, J.J. Hull, Font and Function Word Identifi cation in Document Recognition, *Computer Vision and Image Understanding*, **63**, 1, January 1996, 66-74.

[18] A. Konheim, *Cryptography, a primer*, Wiley, New York 1981.

[19] D.S. Lee, J.J. Hull, Information extraction from symbolically compressed images, *Procs. SDIUT*, Annapolis 1999, 176-182.

[20] D.S. Lee, J.J. Hull, Duplicate detection for symbolically compressed documents, *Procs. ICDAR-5*, Bangalore 1999, 305-308.

[21] G. Nagy, S. Seth, K. Einspahr, T. Meyer, Effi cient algorithms to decode substitution ciphers with application to OCR, *Procs. ICPR 8*, Paris, 1986, 352-354.

[22] G. Nagy, S. Seth, K. Einspahr, Decoding substitution ciphers by means of word matching with application to OCR, *IEEE-Trans. PAMI-9*, 5, Sept. 1987, 710-715.

[23] G. Nagy, S. Seth, M. Viswanathan, DIA, OCR, and the WWW, in *Handbook of Character Recognition and Document Image Analysis* (H. Bunke and P.S.P. Wang, editors), World Scientifi c 1097, 729-754.

[24] S. Peleg and A. Rosenfeld, Breaking substitution ciphers using relaxation algorithm, *Commun. ACM*, **22**, Nov. 1979, 598-605.

[25] S.V. Rice, F.R. Jenkins, T.A. Nartker, The fi fth annual test of OCR accuracy, TR 96-01, Information Science Research Institute, University of Nevada - Las Vegas, April 1996.

[26] C. Shannon, Communication theory of secrecy systems, *Bell System Technical J.*, **28**, 1949, 636-715.

[27] E.A. Williams, *An invitation to cryptograms*, Simon and Schuster, New York 1959.