Stop Word Location and Identification for Adaptive Text Recognition[1]

Tin Kam Ho
Bell Laboratories, Lucent Technologies
700 Mountain Avenue, 2C-425, Murray Hill, NJ 07974, USA
email: tkh@bell-labs.com

Abstract

We propose a new adaptive strategy for text recognition that attempts to derive knowledge about the dominant font on a given page. The strategy uses a linguistic observation that over half of all words in a typical English passage are contained in a small set of less than 150 stop words. A small dictionary of such words are compiled from the Brown corpus. An arbitrary text page first goes through layout analysis that produces word segmentation. A fast procedure is then applied to locate the most likely candidates for those words, using only widths of the word images. The identity of each word is determined using a word shape classifier. Using the word images together with their identities, character prototypes can be extracted using a previously proposed method. We describe experiments using simulated and real images. In an experiment using 400 real page images, we show that on average, 8 distinct characters can be learned from each page, and the method is successful on 90% of all the pages. These can serve as useful seeds to bootstrap font learning.

Keywords: OCR, word recognition, font learning, keyword spotting, adaptive recognition

1 Introduction

Training a symbol classifier for a wide range of shape variations and image degradations has well-known difficulties in machine-print OCR. An alternative strategy is to train the classifier only for the fonts and degradation conditions present in a given text page. Such a strategy is appealing since it can greatly simplify the symbol recognition task, and it is promising for better recognition accuracy. The strategy is practical since in most applications there is usually a dominant font in a given page. It is especially valuable in the cases that multiple pages from the same source need to be processed, as common in many digital library construction tasks.

[1] Parts of this paper have appeared in [7] and [8].

An adaptive OCR strategy requires some knowledge of the dominant font and defects in a given page image. It is often sufficient to represent such knowledge implicitly by character prototypes extracted directly from the image. Some recent attempts on adaptive text recognition [14][17] have suggested good uses of such prototypes and methods for their extraction. In particular, [13] and [14] describe prototype extraction by matching word images to partial ground truth or output from a multi-font OCR.

In this paper, we investigate an alternative strategy that begins by direct recognition of some familiar words in a text page. With conservative and reliable recognition results, one can proceed with character prototype extraction as in [13] and [14] and continue by propagating accumulated knowledge to recognize the rest of the text. The advantage here is that by focusing on a small set of common words, more elaborate and reliable methods [4] can be applied to word recognition, so that the strategy is more robust for variable quality images, such as those sent via fax transmission. In particular, word shape based methods are promising, since the rich structure in a word shape and wider differences between shapes of different words yield an easier symbol recognition problem. Word shapes and function word statistics have been used for font learning in [11] which also demonstrated the feasibility of identifying the dominant font among a known set.

2 A linguistic observation

Although there have been extensive linguistic analyses on frequency distribution of English words, such statistics are not always useful in OCR. One reason is that statistics derived from large corpora, such as letter, n-gram, or word frequencies, could deviate largely from those observed from small samples, like the relatively short text (one page or a few pages) typically handled by OCR devices. Consequently one has be to very careful in applying probabilistic language models to short text.

However, there is one observation that surprisingly holds true for very short text passages: if one makes a list of about 150 most common words from modern American English usage, and takes any random word from an American English sentence, there is a greater than 50%

chance that this word will be contained in the list. This fact is used in a recently proposed method to decode short cryptograms [3].

To see if the same holds true for conditions encountered by OCR devices, we used a dictionary of 134 most frequent words (Figure 1) from the Brown corpus [12] and counted the occurrences of such words using a collection of text samples that are of lengths no more than those typically processed by OCR. For consistency with our emphasis on low-quality images, we chose the fax images of the Business Letter Sample of the UNLV database as our test set. Previous OCR results on this database have been reported in [16]. There are 200 pages in this collection.

```
a about after all also an and any are as at back be
because been before being between both but by can could
day did do down each even first for from get good had has
have he her here him his how I if in into is it its just
know life like little long made make man many may me men
more most Mr much must my never new no not now of old on
one only or other our out over own people said see she
should so some state still such than that the their them
then there these they this those three through time to too
two under up very was way we well were what when where
which who will with work world would year years you your
```

Figure 1: A list of 134 most common words in Brown corpus.

Figure 2 (a) and (b) show the per page count and fraction of words that are contained in this dictionary, where the pages are order by increasing magnitude of these numbers. Except for a few pages containing very short business letters, we found the hypothesis is roughly true: most pages have from 40% to 50% of the words occurring in the dictionary we specified. The slight deviation from the expected 50% is due to many non-sentences (such as dates and address blocks) in business letters. On average, each page has 248.75 words, and about 43% of them are contained in the dictionary. Even in the shortest letter, 12 of the 25 words are found in the dictionary. Thus a strategy based on recognizing these words will be applicable

to every such page.

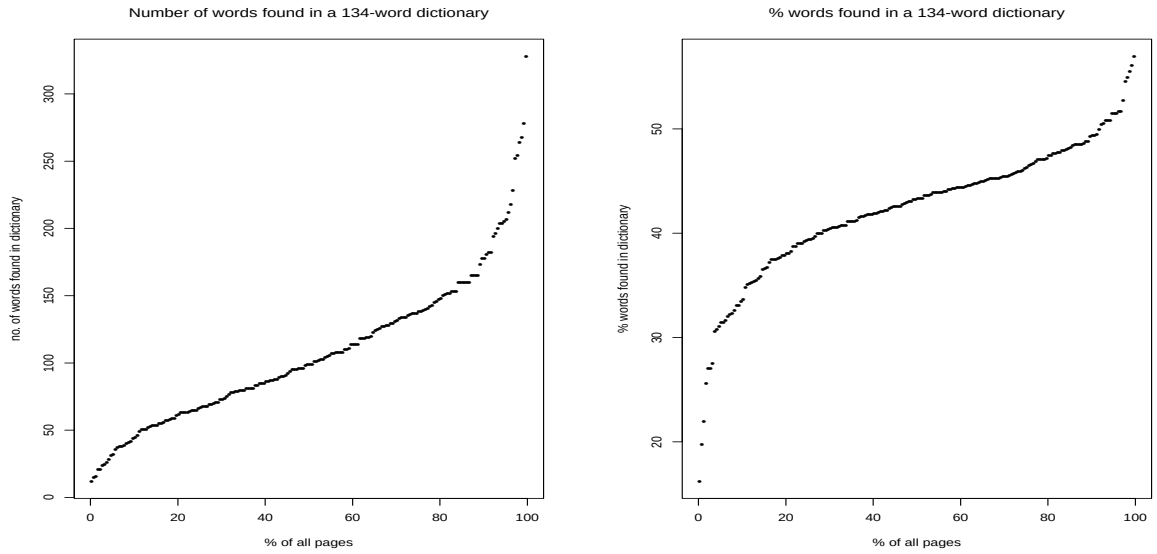


Figure 2: Per page count and fraction of words contained in a 134-word dictionary.

3 Location of stop words on a page

Our adaptive strategy is to recognize first those stop words from a text page. As proposed in [13] and [14], images and identities of these words can then be matched to extract character prototypes for uses in template matching and level building. Since this strategy includes several sequential steps, errors in the initial steps should be kept to a minimum, or their propagation will severely affect final recognition. Therefore we begin with an initial step that uses contextual information to locate the stop words and eliminate non-stop words from consideration. The advantages are twofold. First, early elimination of non-stop words will prevent them from being misidentified as a stop word by shape matching. Second, word shape analysis need only be performed on the more likely candidates, thus computation time can be kept to a minimum.

We explored the use of word width estimates in the context of a sentence to assist location of the stop words. Notice that previous studies [2] have found that width estimates of *single*

words are unreliable for stop word identification. However, since many such stop words are prepositions, we suspect that their contexts may have certain distinguishing characteristics. In particular, we conjectured that the widths of these words together with those of their immediate neighbors may reveal their presences. In this study we investigated this conjecture to see if it leads to a rule for discriminating between stop and non-stop words.

Discrimination between stop words and non-stop words has been an important concern in information retrieval. However, in information retrieval studies, the goal is to discard those identified as stop words so that analysis can be focused on the words conveying more information. In our text recognition strategy the goal is the opposite: we attempt to find the most common stop words and discard the rest, on the expectation that, by our familiarity with the shapes of those common words, we can infer knowledge of symbol shapes that will eventually help us recognize the rest of the text.

While our main goal is to find the stop words, by discriminating between the two types of words, the procedure carries a side effect of finding likely candidates for non-stop words and thus will also be useful for keyword spotting from text images.

We tested the feasibility of the method using simulated data first, and then implemented the procedure for real images. The following sections describe details of the method, the design of the experiments, and the results.

4 Distribution of word width statistics

In ordinary text each word may appear in lower case or with an initial capital letter. For imaged text that makes a difference in the word width. For simplicity, we considered only lower case instances of the most common words, so we excluded the words "I" and "Mr" from our dictionary since they do not typically appear in lower case. Though, in an alternative implementation, they can also be treated as special cases. From now on when we say stop words we refer to only words in the remaining list of 132 words. This change has negligible effect on the proportion of stop words on a page. For instance, among the 200 pages from

the UNLV sample, 83 pages have no changes, and the rest 117 pages have only, on average, 1.5% decrease in stop word proportion.

We collected statistics of word widths from 500 articles in the Brown corpus. We were concerned about the distribution of the statistics conditional on two classes "Stop" (for stop words in our list) and "Non" (for other alphabetical strings). We will refer to the words in both classes collectively as the "candidate strings", with the understanding that for the "Stop" class it refers to any words in the fixed list and for the "Non" class it can be any word outside the list. Numeral or alphanumeric strings were excluded.

We measured the widths of each candidate string and the widths of the words occurring before and after it. We considered only those occurrences of the candidate strings that are in lower case, and that are not the first or last word in any sentence. For each triplet of consecutive words from each sentence, we called the widths of the three words w_1, w_2, and w_3 respectively and used $(w_2, w_1/w_2, w_3/w_2)$ as a feature vector. Distributions of the two classes of words in this three dimensional feature space were studied.

Widths of ASCII strings

First the ASCII representation of the corpus was used. Detection of any fixed word from ASCII text is trivial; here we just used such text as ideal document pages where character width is a constant, so that the effect of the linguistic context can be measured in the absence of imaging effects.

For this representation, the width of each character W_c is taken as 1, and the width of each word is $n \times W_c$ which is simply n, the number of characters in the word. Figure 3 shows the number of occurrences of candidate strings and the stop words among them in each of the 500 articles. The articles are sorted in increasing order of length (number of candidate strings). Notice that the fraction of stop words is generally close to the 50% estimate, confirming the truth of the linguistic assertion. The average percentage over all 500 articles is 52.8% and the standard deviation is 3.86%. The average length of the articles is 1500.5 candidate

strings and the standard deviation is 97.

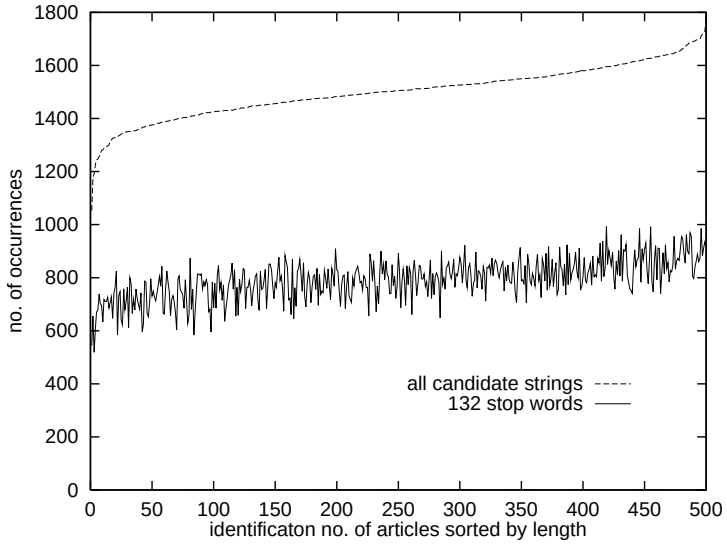


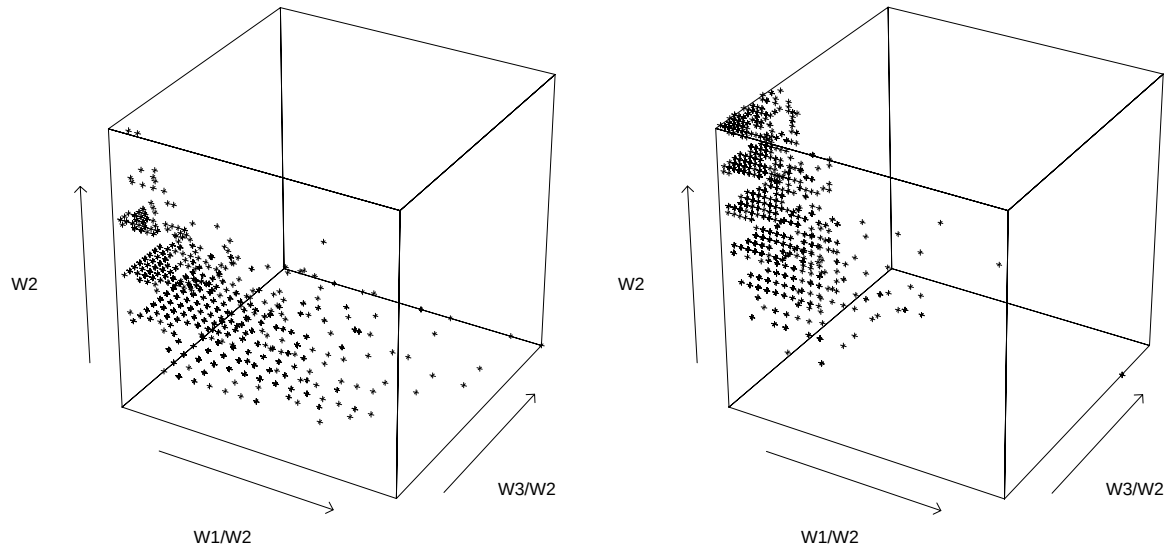
Figure 3: Number of occurrences of candidate strings and stop words in each article.

To measure the discriminative power of the features, we split the article collection into two halves TR and TE. The split was stratified by the content category (press, religion, skills, fiction, etc.) so that articles of similar subjects occupy roughly the same proportion in each half. Within each category all odd numbered articles were assigned to TR and the even numbered to TE. There are 254 articles in TR and 246 in TE. As a result there are 380,255 candidate strings in TR and 370,000 in TE.

Figure 4 shows the distributions of the first 1000 words of each class in the feature space $(x, y, z) = (w_2, w_1/w_2, w_3/w_2)$. As can be seen they are well separated in most regions. The observed distributions have very long tails, in x resulting mostly from rare occurrences of wider words (words with more than 15 characters) and in y, z from division by small widths of some stop words. So we limited the range of each coordinate as follows: $x \in [1, 8], y \in (0, 9], z \in (0, 9]$. All values beyond the upper bounds were replaced by the upper bounds in density estimation.

Since there are only three dimensions in the space, we can afford a binning procedure

to estimate the probability densities. Values in x are integers in this case (just counts of characters in the ASCII word). Values in y and z are binned to their ceilings. Also since all our stop words have 7 or less characters, the distribution is not interesting when $x > 7$. We thus took $7 \times 9 \times 9 = 567$ equally sized bins to estimate the probabilities. We assumed the *a priori* probability of the two classes to be equal following the linguistic assertion. Using articles in TR, estimates of the *a posteriori* probabilities $P(c|(x, y, z))$ of the two classes ($c = Stop$ or Non) in each bin show large differences especially near the extreme ends of the x range.



1000 stop words 1000 nonstop words

Figure 4: Distribution of stop and nonstop words in the feature space $(w_2, w_1/w_2, w_3/w_2)$ within the chosen limits as observed using ASCII text.

Based on these estimates we designed a Bayes classifier with a reject option. A high reject threshold can be chosen so that decision is avoided in the regions where the difference between probabilities of the two classes is small. The classifier uses the following rule (with

$thres \geq 0.5$):

$$\text{decide } Stop \text{ if } P(Stop|(x,y,z)) \geq thres,$$

$$\text{else} \quad \text{decide } Non \text{ if } P(Non|(x,y,z)) \geq thres,$$

$$\text{and} \quad \text{reject otherwise.}$$

Applying this classifier to all candidate strings in TE, we obtained accuracy/reject tradeoff as shown in Table 1. From the table we can see that, using a threshold of 0.99, highly reliable ($> 99\%$ in confidence, column (d)) recognition of stop words can be achieved. This occurs for 4.45% of all candidate strings (column (a) $\times$ column (c)), i.e., on average 67 stop words per article can be identified this way based on an expected article length of 1500 candidate strings. Notice that this did not involve any detailed word shape analysis. 92% of those easily identifiable stop words are one of the following ten words, in the order of descending relative frequency (shown in parentheses):

```
of (32.72%) to (17.51%) in (15.22%) by (5.24%) on (4.59%)
 a ( 4.57%) is ( 4.01%) at ( 3.29%) as (2.79%) or (2.00%)
```

It is interesting to note that except for "a", all these words are composed of two letters. In fact, using a threshold of 0.99, 97.99% of the identified stop words have two letters (1.62% have one, 0.32% have three, and 0.07% have four letters respectively). Using a threshold of 0.95, the distribution becomes 84.17% for words with two letters, 4.46% one letter, 11.35% three letters, and 0.02% four letters respectively. The followings are some example word triplets that contain one of the easily detectable stop words. It is evident that in most of these cases there is a short stop word surrounded by two longer words.

```
fail to convey / going to get / inverse is the / going to determine /
and the business-minded / nature of the / could be doing / as a whole /
leads to the / vision on the / from a converted / for by profit-sharing /
```

```
expected to know / full on the / need to maintain / fifteen to twenty /
dog in the / tightened up in / almost an obsession / formation of your /
everyone is talking / violation of the / mounted by means / were on the /
strengthen our rational / matter how determined / coldness and affection /
field are expandable / there was brilliance / domestic and international /
twelve-year-old with child-bearing / spaces over non-algebraically /
In this relationship-building / ...
```

Table 1: Accuracy and reject tradeoff of stop word decision based on word width statistics. (a):decided / all candidate strings; (b): correct/all decided; (c): decided as stop word / all decided; (d): correct/all decided as stop word; (e): decided as non-stop word/ all decided; (f): correct / all decided as non-stop word.

	all candidate strings		decided as stop words		decided as non-stop words	
thres	% decided	% correct	% decided	% correct	% decided	% correct
	(a)	(b)	(c)	(d)	(e)	(f)
0.99	19.56	99.78	22.76	99.02	77.24	100.00
0.98	27.05	99.34	40.81	98.50	59.19	99.92
0.97	31.44	99.07	49.06	98.18	50.94	99.92
0.96	40.84	98.50	44.33	97.92	55.67	98.96
0.95	48.87	97.97	41.77	97.56	58.23	98.26
0.90	66.66	96.36	54.43	95.04	45.57	97.93
0.85	70.81	95.76	56.92	94.15	43.08	97.89
0.80	71.45	95.63	57.30	93.94	42.70	97.89
0.75	77.62	94.07	53.28	93.76	46.72	94.43
0.70	83.19	92.56	54.29	91.89	45.71	93.35
0.65	86.86	91.52	53.81	91.05	46.19	92.06
0.60	90.55	90.40	55.06	89.38	44.94	91.66
0.55	93.88	89.21	56.37	87.46	43.63	91.46
0.50	99.99	86.94	52.93	87.46	47.07	86.36

Using a threshold of 0.95, decisions can be made on 49% of all candidate strings with a 98% accuracy. Another result that can be read from Table 1 is that, using a threshold of 0.50, we can eliminate 47% (column (a) $\times$ column (e)) of the candidate strings from consideration as stop words, while risking a loss of 12% ((47% $\times$ (100% - column (f)))/50%) of the real

stop words. The nonzero reject rate under this threshold is due to certain feature values not occurring in TR so that both probability estimates are zero.

Since font learning does not have to use every stop word on a page, we prefer a high threshold which gives more reliable decisions but may leave many stop words undetected. However, if the discriminator is to be optimized for keyword spotting applications, a lower threshold will be preferred.

From this analysis we can see that, for idealized, noiseless text pages in a single font, a very quick and reliable discrimination can be made between stop words and non-stop words using purely word width information. This discrimination will enable us to focus on a small number of words that are highly likely to be stop words, and discard many words that are unlikely so. Shape matching for font learning can then begin with the focus words, and proceed to use the rejected words only if needed. With this fast procedure we can avoid costly shape matching for all words on a given page. In the following sections we see what happens when the font type is uncertain and when imaging noise comes in.

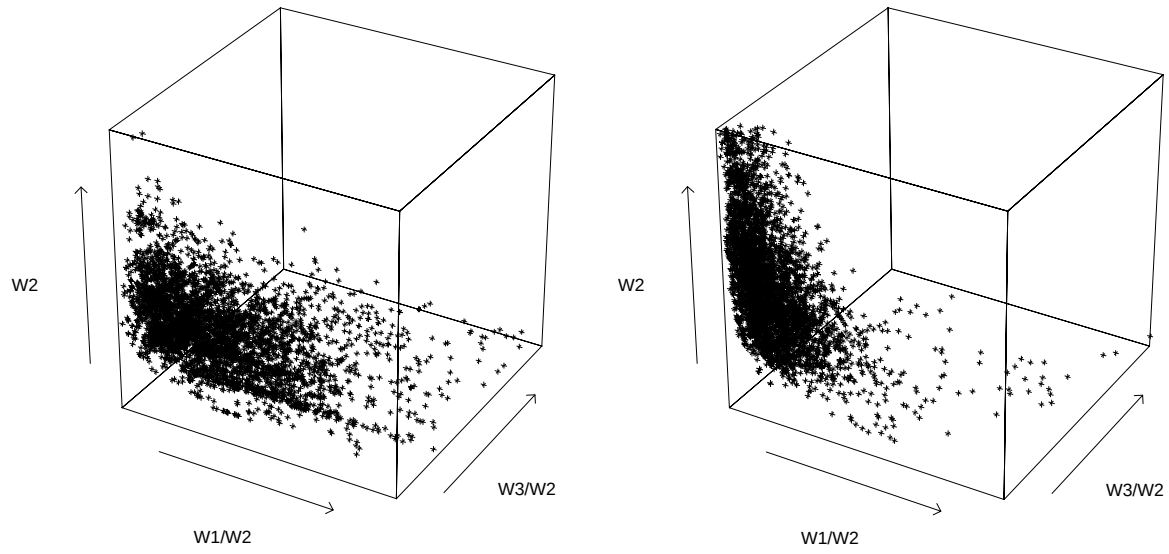
Widths of simulated word images

The estimates and tests on ASCII text assume a constant character width of 1 and that all word widths are simple multiples of character widths. While this is a simplifying assumption, it is not too far from truth for clean images of text printed in fixed pitch fonts. In this section we describe the same analysis using simulated images of text in several typical fonts.

To account for differences between fonts, we measured the character widths relative to the capheight of each font. The reason is that capheight estimation is relatively easy for real images once a baseline finding procedure is applied to every located text line. Note that this normalization is good only as long as the aspect ratios do not differ by much between the chosen fonts, otherwise, values of x for the same word will scatter over a large range. Here values of x are no more integers after this normalization, and they are binned to their ceilings as well. In the experiment we used four popular fonts: "Adobe Times-Roman" (R),

"Adobe Helvetica Roman" (H), "Bitstream Courier Roman" (cr), and "Computer Modern Roman" (tR). We did not consider boldfaces or italics.

To avoid complications due to variable inter-character spacing, we assumed that all vertical gaps between characters within a word were removed. This enabled us to calculate the width of a word as a simple sum of widths of its component characters. For degraded images, widths of words may be larger due to blurring of the stroke edges or smaller due to faint and narrowed strokes. Though, if we assume degradation is even across a local region on a page, we expect at least some differences be canceled out when we take ratios of the widths. Figure 5 shows the resulting distributions. We can see that the distributions largely resemble those derived from ASCII text.



1000 stop words 1000 nonstop words

Figure 5: Distribution of stop and nonstop words in the feature space $(w_2, w_1/w_2, w_3/w_2)$ within the chosen limits as observed using simulated images in four fonts.

Using articles in TR, we constructed a similar Bayes classifier using character widths of the four font samples (R, H, cr, tR). Table 2 shows the performance of this classifier on TE, also simulated using the same four fonts.

Table 2: Accuracy and reject tradeoff of stop word decision based on word width statistics. (a):decided / all candidate strings; (b): correct/all decided; (c): decided as stop word / all decided; (d): correct/all decided as stop word; (e): decided as non-stop word/ all decided; (f): correct / all decided as non-stop word.

	all candidate strings		decided as stop words		decided as non-stop words	
thres	% decided	% correct	% decided	% correct	% decided	% correct
	(a)	(b)	(c)	(d)	(e)	(f)
0.99	9.30	99.38	10.51	98.89	89.49	99.44
0.98	15.26	99.09	27.19	98.48	72.81	99.32
0.97	17.35	98.88	35.54	98.11	64.46	99.31
0.96	22.90	98.22	49.04	97.18	50.96	99.21
0.95	26.85	97.76	56.53	96.64	43.47	99.21
0.90	43.97	95.90	56.83	94.87	43.17	97.25
0.85	61.21	93.37	53.57	93.12	46.43	93.66
0.80	67.80	92.29	56.30	91.50	43.70	93.30
0.75	72.59	91.31	55.16	90.80	44.84	91.94
0.70	76.80	90.28	57.62	89.05	42.38	91.94
0.65	79.37	89.58	55.94	88.97	44.06	90.36
0.60	87.07	87.20	51.98	88.50	48.02	85.80
0.55	97.82	83.81	56.37	82.73	43.63	85.20
0.50	98.40	83.63	56.33	82.57	43.67	84.99

From Table 2 we see slight degradation from ASCII-based results. However, generally high reliability is maintained in the discrimination. Say, using a threshold of 0.90, decisions can be made on 44% of all candidate words with a 96% accuracy, and of those decided as stop words 95% are correct. Using a threshold of 0.50, 43% of words can be eliminated from consideration as stop words with a risk of losing 13% of the true ones. A graphical summary of the tradeoffs is given in Figure 6.

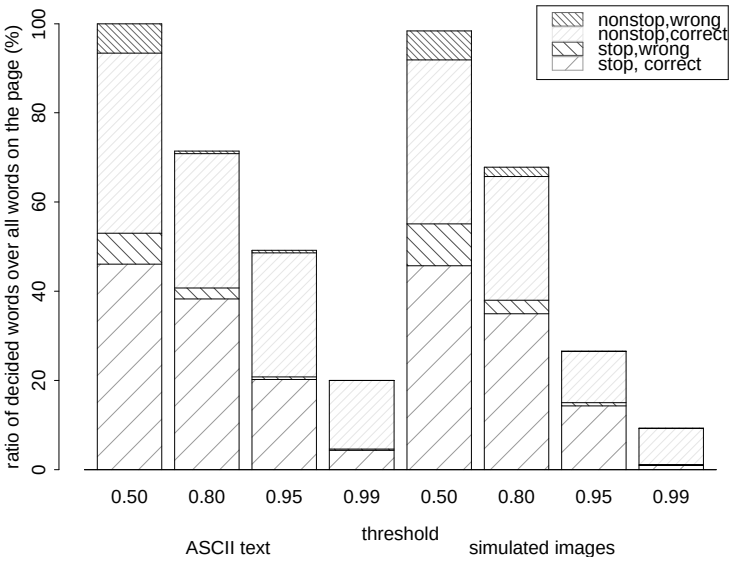


Figure 6: Accuracy and reject tradeoff of stop word decisions.

5 Stop word identification for real text images

The experiments using ASCII text or simulated images suggest that the feature vector $(x, y, z) = (w_2, w_1/w_2, w_3/w_2)$ is effective in discriminating between the two types of words. However, such experiments assume that the width of each character is a fixed number or one of several fixed numbers. When measurements are taken from real text images, this assumption is usually not valid. First, the text may appear in a font not included in the training data. Second, even if the font is expected, distortions and spatial quantization introduced during printing and scanning may yield variable widths for the same symbol within the same page. Additional difficulties in measuring the word widths include errors in segmenting the page and identifying text in the dominant font. So the performance of the classifier is questionable. In this section we describe an experiment on several sets of imaged text.

To apply the discrimination rule to imaged text, the page image was first segmented to word level. For this we used the layout analysis module in the Murray Hill Page Reader [10], together with simple rules to merge neighboring characters to form words. Then the median

capheight of all detected text lines was obtained for normalization of word widths. All text lines within a small variation of the median capheight were assumed to be in the dominant font. We then computed the feature vector (x, y, z) for each detected word triplet, and to each vector applied the classifier derived using simulated images of four fonts (as described in the previous section). A threshold of 0.8 was used in the decision rule.

Notice that in this process, sentence breaks were not accounted for because we did not include a procedure for sentence segmentation. Rather, each triplet of consecutive words in a block of text was classified. Omission of sentence breaks introduced errors due to different occurrence frequencies of some words around sentence breaks or within the sentences and differences in widths of capital letters from those of lower case letters at the beginning of each sentence.

Three datasets containing a total of 400 text pages were used in the experiment. Among those, 200 were pages from a diverse collection of technical journals, and 200 pages were samples of business letters. The journal pages included 100 pages from our local database (400x400 dpi, referred to as **J**) and 100 pages from the UW English Document Image Database I ("A001" to "A061", 300x300 dpi, referred to as **UW**) [15]. The letter samples were fine-mode fax images from the English Business Letter Sample of the UNLV database (204x196 dpi, referred to as **B**) [16]. Thus the dataset covered three different resolutions, at least three different scanners, two types of contents and layout formats, and many different document sources.

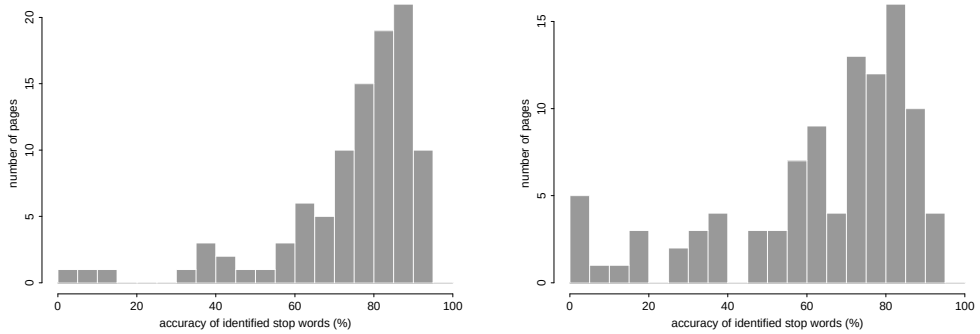
Figure 7 shows the accuracy of identified stop words using the classifier for the two sets of journal pages separately, and Figure 8 compares their totals to those of the letter samples. The accuracy shown was the count of verified stop words over the count of words decided to be a stop word according to the classifier. For a summary statistic, the median accuracy for set **J** is 80%, for set **UW** is 72%, and for set **B** is 72%. The third quartile for set **J** is 86%, for set **UW** is 82%, for set **B** is 80%. It should be noted that differences in accuracy between these three datasets include mixed effects of document sources, printer and scanner

properties, scanning resolution, layout formats, and types of contents.

Notice that, according to the linguistic assertion, a trivial classifier that decides randomly would be at an accuracy close to 50%. From Figure 9 we can see that the mode of accuracy distribution is near 80%. Due to a long tail on the left, the average accuracy of our classifier over the entire collection of 400 images is 68.79%, which gives a 95% confidence interval of (64.08%, 73.14%), significantly different from triviality. The median (75%) and third quartile (83%) accuracies show that for a substantial portion of the images the stop word decisions are reliable. However, on occasions the accuracy actually falls below 50%. A couple of those at zero accuracy (1 page for **J**, 2 for **UW**, and 1 for **B**) were due to failure of page segmentation. Those very close to zero were mostly due to mistakes in finding text in the dominant font, e.g., cases where a table occupies all or most of the page area. In less extreme cases, errors were caused by an assortment of reasons:

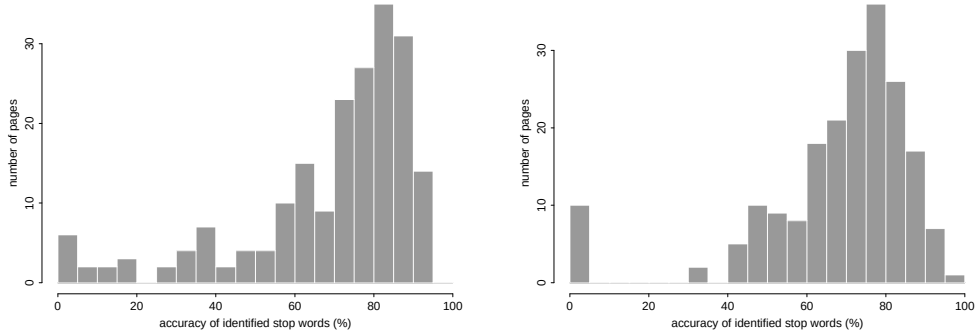
- tables or mathematical formulas misidentified as text blocks;
- failure in separating text blocks from graphics (e.g. logos);
- special text blocks such as a bibliographical list;
- errors in detecting word breaks and hyphenated words across two lines;
- punctuations (e.g.,"&", "%"), numeral and other non-alpha strings (e.g., dates and dollar amounts) treated as words;
- proper nouns, acronyms, and abbreviations such as "e.g.", "i.e.";
- mixed-font text lines (smallcaps, boldfaces, and italics);
- non-English text (e.g. French titles in **UW** page "A041");
- 90 degree rotated text (e.g. **UW** page "A002");
- blobs of dirt treated as words.

Note that these are in addition to expected errors from omission of sentence breaks, variations in widths of imaged symbol shapes of known fonts, and unexpected widths of unseen



(a) (b)

Figure 7: Accuracy of identified stop words for (a) 100 journal pages in 400x400dpi; (b) 100 journal pages in 300x300 dpi.



(a) (b)

Figure 8: Accuracy of identified stop words for (a) 200 journal pages in 400 or 300 dpi; (b) 200 business letters in 204x196 dpi.

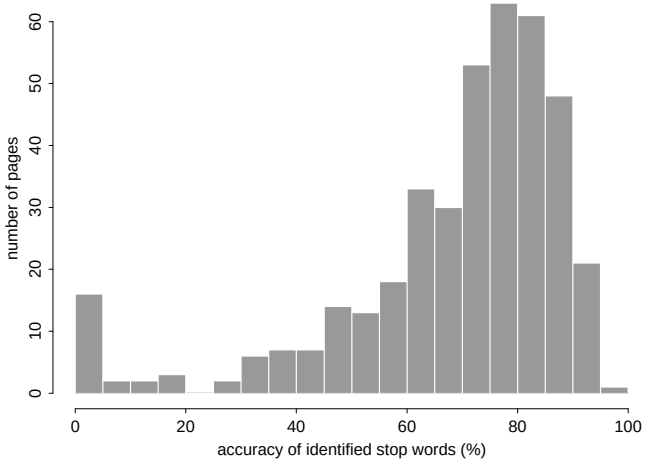


Figure 9: Accuracy of identified stop words for all 400 pages.

font types. Among all, the most notable effects were those of deviation of text contents from the news articles that were used to train the classifier, since comparing to ordinary text, technical journal pages contain far more numeral strings, mathematical formulas, and bibliographical citations, and letter samples typically contain dates, address blocks, person names, but only short passages of regular text.

Therefore while we believe that the features are effective and the classifier is useful, we suggest that the method be coupled carefully with detailed layout analysis that includes simple logical labeling of detected text blocks. Such logical labeling will, for instances, prevent the classifier from being applied to words found in a table or an address block. Retraining the classifier with more font types and text of expected types of contents should also improve the results. Training data can be further enhanced by simulated degradations of the font samples [1]. Ideally, the classifier should be trained on words resulting from segmentation of real page images rather than synthetic ones.

Since the method is intended to be a first step of an adaptive recognition strategy, it is essential to know if this method can detect enough stop words for font learning. The results show that on average, 146 *correct* stop words are obtained for set **J**, 100 for set **UW**, and 49 for set **B** where there is less regular text on each page. These are still sufficient to support a conservative shape-based recognition method in the next step.

Figure 10 shows the first 50 stop words identified from a page in set **J**, including some errors (e.g. "chal-"). It can be seen that a diverse set of words are included. Note that the inclusion of the word "The" is accidental, since the three consecutive words centered at it spread across a sentence break. To remove the errors, detailed shape matching will need to be invoked.

Using this method, on average, only 37% of all word images on a page need to be passed to subsequent shape analysis (39% for set **J**, 41% for set **UW**, and 34% for set **B**). This is a substantial improvement on efficiency. Since layout analysis is part of the recognition strategy in any case, the only additional procedures involved in this method are white column

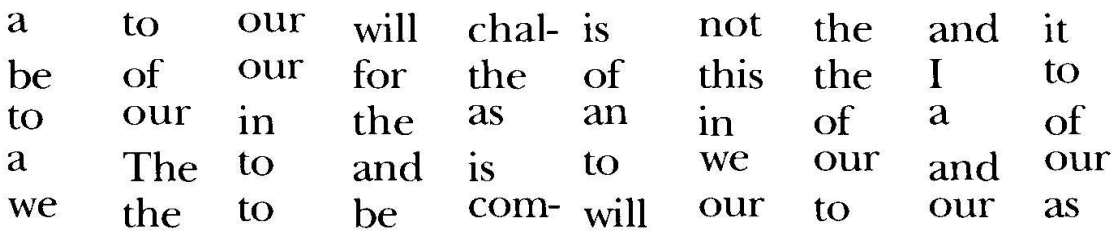


Figure 10: Examples of identified stop words including errors.

removal for each word, word width calculations, and application of the very simple decision rule.

6 Recognition of located stop words

To recognize the identity of each stop word located by the above procedure, we used a "holistic" technique that does not depend on reliable character segmentation and recognition which is the challenge for low-quality images in the first place. Holistic techniques treat a word as a single symbol, and recognition is by matching the whole shape of a word image to pre-fabricated prototypes [4].

The same four fonts (R, H, cr, tR) were used to create the word prototypes. First, synthetic images of words in the stop word lexicon were composed using font prototypes according to usual typographic rules. Degraded images of the synthetic words were then generated pseudo-randomly using a document image defect model [1]. The procedure was repeated for each word and each font. Altogether, 400 samples were created for each word, representing 4 fonts, 4 point sizes (8,10,12,14), and 25 tuples of pseudo-random defect parameters. The output resolution was set at 200x200 to simulate (fine mode) fax quality. To extract image features, each word image was size-normalized to 16 rows $\times$ 64 columns. All white gaps between characters were removed before size normalization. Figure 11 shows some size-normalized synthetic word samples.

The feature vector is the concatenation of four vectors: (1) a "binary subsamples" vector, (2) a "pixel correlation" vector, (3) a "vertical runs count" vector, and (4) a "horizontal runs count" vector.

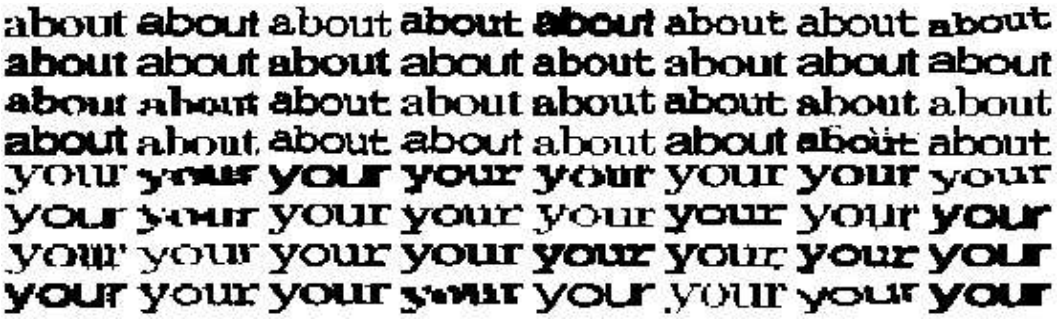


Figure 11: Synthetic samples of two words created using the defect model.

To compute the subsamples, a word image was scanned from left to right and top to bottom; in the process each 2×2 (nonoverlapping) region was represented by 1 if three or more pixels were black and 0 otherwise. The same process was then repeated for the reduced image until there was only one remaining row or column. The pixel correlation features were conjunctions and disjunctions of neighboring pixels in various directions, and are detailed in [5]. The other two vectors were simply counts of black runs (contiguous blocks of black pixels) in each column and row of the normalized image.

The feature vectors describing the synthetic images were then used to construct a decision forest [6]. The choice of a decision forest as the classifier was motivated by its accuracy, speed, and that its natural confidence measure can be thresholded to identify reliable decisions. For better resolution of the confidence measure, similar word samples were created for three more fonts: ITC Bookman Medium Roman, Computer Modern Sans Serif, and Computer Modern Typewriter. Samples of these fonts were used to populate the forest after construction. All words were synthesized only in lower case.

A highly conservative threshold was set ($c \geq 0.9$) on the confidence of the recognition decisions, so that all word images recognized with less than 0.9 confidence were rejected as unreliable. Using this threshold, all accepted decisions are correct. Table 3 lists the number of stop words identified and those having reliable decisions for each test set. Layout analysis failed for one page in each of **J** and **UW**. Other than that, for about 10% of the pages, there are no accepted decisions. Of the 36 such cases, half of them have few (less than 50)

stop words identified on the page, so that under our very conservative confidence threshold no recognition decisions were considered reliable. For the rest, failure was due to either problematic word segmentation, where many single characters showed up as stop words, or uncommon font types for which word shape recognition was not able to handle with high enough confidence.

Among the 132 words used, only about 20 of them are highly frequent. The most commonly recognized words are `to, a, and, of` (Table 4). Some obviously content-dependent cases are the words `you, your, my` which showed up very often in the business letter sample but not in the technical journals. No reliable recognition occurred for the following 56 words in the 132-word dictionary: `about, after, back, because, been, before, between, both, could, down, even, first, good, her, here, him, how, just, know, life, like, little, long, made, make, man, many, may, men, most, much, must, never, only, other, over, people, should, state, their, them, there, those, three, through, under, way, well, what, where, which, who, work, world, would, year.` This suggests that speed and possibly accuracy can be further improved by restricting to a much smaller stop word dictionary.

Averaged over all three data sets, 8 distinct characters (most with multiple samples) can be obtained from each successfully processed page. These can serve well as seeds for a bootstrapping strategy. A strategy proposed by Nagy and Xu [13][14] is a natural continuation from here. Their method takes as input word images with identities and look for word pairs with identical characters. Those images are compared to extract the common parts. Character images isolated this way can be used to start a recursive segmentation process. In this process, word images are first segmented at the position where a character has been extracted, with the decision string divided accordingly. Then characters obtained from other words are shifted on top of the partial word images for further matching and segmentation. The process continues until no prototypes are available to match the remaining partial images [9].

Table 3: Results of stop word recognition.

Set	#pages	#stop words found	#accepted recognition	#recognized words per page	#distinct chars per page	#pages w/o accepted recognition
B	200	14377	2104	10.5	7.3	19
J	99	19699	2827	28.5	10.3	5
UW	99	14141	1475	14.9	8.3	12
total	398	48217	6406	16.1	8.3	36

7 Conclusions

We presented a method for recognizing the most common stop words from a page image for the purpose of extracting character prototypes. After layout analysis producing word images, a fast estimate is made on the widths of each word and its immediate neighbors. The width ratios are used in a Bayes classifier to decide if the word is a candidate stop word. A decision forest classifier is then applied to determine its identity. We have shown that, on a test set of 400 images of text pages covering two types of contents and three different resolutions, stop words can be reliably recognized from 90% of the pages that yield, on average, 8 distinct characters. We believe this method can be a useful start for an adaptive strategy.

In our experiments we have used the method only for American English texts. For the method to be applicable to other languages, it is necessary that similar linguistic characteristics exist. In particular, the method depends on facts like that a small set of frequently occurring words are available, and that they are detectable by comparing lengths of surrounding words. Nevertheless, we expect these to be true for many other Latin based languages.

While this work has enabled learning of some character shapes specific to a page, it still relies on a broad enough collection of font samples to train the stop word classifier. That means it will not be able to handle some exotic shapes. This is a constraint imposed by automatic shape classification as compared to a strategy based on human intervention (e.g., [13] [14]). However, this is true for all other adaptation strategies that depend on shape

Table 4: Frequency of recognition by word and by dataset, sorted by total frequency.

word	**B**	**J**	**UW**	total	word	**B**	**J**	**UW**	total
a	375	580	417	1372	this	4	6	0	10
of	176	462	221	859	did	1	2	5	8
to	385	315	134	834	with	5	1	1	7
and	227	229	109	565	one	1	4	2	7
for	112	100	59	271	from	1	4	2	7
by	64	90	78	232	will	1	2	3	6
at	59	105	56	220	two	1	3	2	6
are	39	119	57	215	have	1	3	2	6
you	162	5	0	167	each	0	4	2	6
an	22	100	40	162	old	0	4	1	5
we	47	64	21	132	new	3	1	1	5
on	35	61	23	119	get	4	1	0	5
the	19	76	20	115	very	2	1	1	4
as	21	52	41	114	own	2	2	0	4
is	20	81	11	112	than	0	3	0	3
it	44	34	17	95	its	0	3	0	3
or	34	39	6	79	but	1	2	0	3
our	44	10	2	56	were	1	0	1	2
was	0	22	32	54	time	0	2	0	2
so	11	30	8	49	then	1	1	0	2
do	20	18	11	49	still	0	2	0	2
can	4	31	7	42	see	1	1	0	2
no	13	11	12	36	said	0	2	0	2
any	25	6	3	34	out	0	1	1	2
has	8	18	5	31	me	2	0	0	2
be	5	7	18	30	years	0	1	0	1
your	29	0	0	29	when	0	1	0	1
into	6	13	10	29	too	0	0	1	1
in	17	10	2	29	they	1	0	0	1
not	7	15	2	24	these	0	1	0	1
up	10	9	2	21	such	0	1	0	1
that	6	6	7	19	some	0	1	0	1
if	6	8	3	17	she	0	1	0	1
my	14	0	0	14	now	1	0	0	1
he	1	11	2	14	more	1	0	0	1
had	0	11	3	14	his	0	1	0	1
also	0	8	6	14	day	1	0	0	1
all	1	8	5	14	being	0	1	0	1

recognition in any way (e.g., [17]). And strictly speaking, even a human is limited by his/her ability to remember and generalize previously learned symbol shapes. A completely adaptive strategy will need to be driven entirely by data from the page, i.e., through an unsupervised learning procedure.

8 Acknowledgements

The author would like to thank Henry Baird, George Nagy, Jonathan Hull and the anonymous reviewers for many helpful comments. Also, thanks to YiHong Xu who shared her word recognition software that demonstrates the feasibility of the character prototype extraction procedure.

References

[1] H.S. Baird, "Document Image Defect Models", in H.S. Baird, H. Bunke, K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag, pp. 546-556, 1992.

[2] F.R. Chen, D.S. Bloomberg, Extraction of Thematically Relevant Text, *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, April 1996, 163-178.

[3] G. Hart, To Decode Short Cryptograms, *Communications of the ACM*, **37**, 9, September 1994, 102-108.

[4] T.K. Ho, J.J. Hull, S.N. Srihari, A Computational Model for Recognition of Multifont Word Images, *Machine Vision and Applications*, **5**, 1992, 157-168.

[5] T.K. Ho, Random Decision Forests, *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, August 14-18, 1995, 278-282.

[6] T.K. Ho, The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 8, August 1998, 832-844.

[7] T.K. Ho, Bootstrapping Text Recognition from Stop Words, *Proceedings of the 14th International Conference on Pattern Recognition*, Brisbane, Australia, August 17-20, 1998, 605-609.

[8] T.K. Ho, Fast Identification of Stop Words for Font Learning and Keyword Spotting, *Proceedings of the 5th International Conference on Document Analysis and Recognition*, Bangalore, India, September 20-22, 1999, 333-336.

[9] T. Hong, *Degraded Text Recognition Using Visual and Linguistic Context*, Doctoral Dissertation, Dept. of Computer Science, SUNY at Buffalo, 1995.

[10] D.J. Ittner, H.S. Baird, Language-Free Layout Analysis, *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, Oct 1993, 336-340.

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

[12] H. Kucera, W.N. Francis, *Computational Analysis of Present-Day American English*, Brown University Press, Providence, R.I., 1967.

[13] G. Nagy, Y. Xu, Priming the Recognizer, *Proceedings of the IAPR Workshop on Document Analysis Systems*, Malvern, PA, October 14-16, 1996, 263-281.

[14] G. Nagy, Y. Xu, Automatic Prototype Extraction for Adaptive OCR, *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, Ulm, Germany, August 18-20, 1997, 278-282.

[15] I.T. Phillips, S. Chen, J. Ha, R.M. Haralick, English Document Database Design and Implementation Methodology, *Proceedings of the 2nd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, April 26-28, 1993, 65-104.

[16] S.V. Rice, F.R. Jenkins, T.A. Nartker, The Fifth Annual Test of OCR Accuracy, Technical Report 96-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1996.

[17] A.L. Spitz, An OCR Based on Character Shape Codes and Lexical Information, *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, August 14-18, 1995, 723-728.