Classification and distribution of optical character recognition errors

Jeffrey Esakov, Daniel P. Lopresti, and Jonathan S. Sandberg

Matsushita Information Technology Laboratory
Two Research Way
Princeton, NJ 08540

ABSTRACT

This paper describes an approach for classifying OCR errors based on a new variation of a well-known dynamic programming algorithm. We present results from a large-scale experiment we performed involving the printing, scanning, and OCR'ing of over one million characters in each of three fonts, Times, Helvetica, and Courier. Our data allows us to draw a number of interesting conclusions about the nature of OCR errors for a particular font, as well as the relationship between error sets for different fonts.

1. INTRODUCTION

Fully automated OCR promises a tremendous cost savings over the current labor-intensive process, yet 100% accuracy is likely to remain an elusive goal for at least the foreseeable future. Aged books, noisy multi-generation photocopies, and faxes, non-standard text layouts, damaged originals, and handwritten mark-up are topics for advanced research. Even in the case of a "clean" source document, problems such as accounting for white space and disambiguating between nearly-identical characters seem fundamentally hard.

A large amount of research has been directed towards improving OCR accuracy after the fact. These post-processing systems take raw text as their input and attempt to identify and correct as many errors as possible. The target could be as modest as correcting keywords for indexing applications, or as ambitious as returning perfect text. A wide variety of techniques have been proposed, including numerous forms of dictionary look-up and spelling correction[1,2] and our own certifiable optical character recognition.[3] Whatever the approach, it is clear that an understanding of the errors that might arise is important to the design of such systems.

Consider the following example:

Original text `The quick brown fox jumps over the lazy dog.`

OCR text `'lhe q~ick brown foxjurnps ovcr tb l azy dog.`

The 44 character input line was subjected to a number of errors during the OCR process, resulting in the 45 character output line. Intuitively, we can classify these as: simple substitutions (e → c), improper segmentation, or *multi-substitutions* (T → 'l, m → rn, he → b), deletions and insertions (in this case both involving a space), and unrecognized characters (u → ~).

While examining simple cases like this is instructive, OCR is obviously a highly complex process that can "break" in many different ways. We cannot hope to draw general conclusions about error behavior by studying small amounts of data in an ad hoc, informal manner. There are many intriguing questions we might ask if it were possible to analyze arbitrarily large amounts of OCR output. After a period of time, would we stop seeing new kinds of errors? How quickly would this happen? How large would the characteristic error set be for a given font and OCR package? Is this set small relative to the total number of possible errors? Do different fonts have different characteristic sets? Questions such as these can only be answered by processing large amounts of OCR data.

Previous large-scale OCR experiments have used real-world documents drawn from mixed sources and qualities to study the complexity of error correction for document indexing[4] and the correlation of errors between different OCR packages.[5] Our aims are somewhat different. To focus on character recognition errors (as opposed to format- or noise-induced errors), we started with a "clean" source document for which we possess a canonical representation: Herman Melville's classic novel, *Moby-Dick*. This adds an important degree of uniformity, controliability, and reproducibility to our studies. We treat OCR as a "black box" and concentrate solely on the errors it generates.

Our approach to classifying OCR errors is presented in Section 2. In Section 3, we describe the procedure we followed in performing a large-scale experiment: printing, scanning, and OCR'ing several different complete copies of *Moby-Dick*. Section 4 discusses the results of this study. Finally, we offer our conclusions in Section 5.

0-8194-1476-X/94/$6.00

2. AN ALGORITHM FOR CLASSIFYING OCR ERRORS

While manual approaches to OCR error classification have been used in the past (e.g., to examine the cost of raising a document's accuracy above a predetermined threshold), gathering the massive amounts of data necessary to draw more general conclusions requires a fully automatic classification procedure. For this we need a model, ideally one that closely mimics the way human experts classify errors.

To address this issue, we turn to the concept of approximate string matching.[6] Other researchers have adopted this formalism as well,[7] but our application is different in that we use the edit model as a general way of classifying OCR errors, not as a tool for evaluating candidate corrections for a particular error.

The relationship between two similar but not necessarily identical strings is made mathematically precise under the edit model. In the traditional case, the following basic operations are permitted:

(*a*) deletion of a character,

(*b*) insertion of a character,

(*c*) substitution of one character for another.

Each of these is assigned a cost, c_{del}, c_{ins}, and c_{sub}, and the minimum cost sequence of operations that transforms one string into the other is called the *edit distance*. This optimization problem can be solved using a well-known dynamic programming algorithm. Let $s_1 s_2 ... s_i$ be the first i characters of the source (original) string, and $t_1 t_2 ... t_j$ be the first j characters of the target (OCR) string. Define d_{ij} to be the distance between the two substrings. The dynamic programming recurrence is:

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + c_{del}(s_i) \\ d_{i,j-1} + c_{ins}(t_j) \\ d_{i-1,j-1} + c_{sub}(s_i, t_j) \end{cases} \quad (1)$$

If, in addition, we also record the choices that lead to the minimums above (the optimal decisions), the resulting trace-back table provides us with a sequence of operations that perform the transformation in question. For the OCR problem, these edits can be equated with the errors in the OCR string.

For our purposes, we must modify Equation 1 to account for other types of basic "edits" specific to OCR. In particular, we have seen the following occur in practice:

(*d*) substitution of two characters for one (1:2 substitution),

(*e*) substitution of one character for two (2:1 substitution),

(*f*) substitution of two characters for two others (2:2 substitution).

With this in mind, our new recurrence becomes:

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + c_{del}(s_i) \\ d_{i,j-1} + c_{ins}(t_j) \\ d_{i-1,j-1} + c_{sub1:1}(s_i, t_j) \\ d_{i-1,j-2} + c_{sub1:2}(s_i, t_{j-1} t_j) \\ d_{i-2,j-1} + c_{sub2:1}(s_{i-1} s_i, t_j) \\ d_{i-2,j-2} + c_{sub2:2}(s_{i-1} s_i, t_{j-1} t_j) \end{cases} \quad (2)$$

This formulation is broad enough to encompass the vast majority of OCR errors we have observed. The extension to more complex substitutions, for example 3:1 and 3:2, is straightforward, but not necessarily instructive. It has been our experience that such errors occur only very rarely (accounting for fewer than 1% of total errors), and extending the model in this way introduces more ambiguity, as will be discussed in a moment.

We note that the cost functions in Equations 1 and 2 are parameterized. In most applications of the edit model, this is just a mathematical formalism – it is almost always the case that all deletions are charged the same cost, for instance. However, to accurately model the OCR process, we make the observation that "white space is hard" and adjust our cost functions accordingly. No human expert, let alone computer, can be expected to know that the indentation we used when enumerating the editing operations (*a*) through (*f*) above is due to a tab stop set at 1/4 inch and not, for example, the result of 6 (or 5 or 7 or ...) "normal" spaces. Hence, we charge less for space deletions/insertions than we do for non-space errors. Similarly, there are other constraints

the cost functions must obey for the sake of consistency (e.g., the cost of a 1:2 substitution should be no greater than the sum of the costs of a 1:1 substitution and an insertion). For the experiment to be described in the next section, we found that the following cost assignments worked well:

$$c_{del}(space) = c_{ins}(space) = 1$$

$$c_{del}(non\text{-}space) = c_{ins}(non\text{-}space) = 3$$

$$c_{sub1:1} = 4$$

$$c_{sub1:2} = c_{sub2:1} = c_{sub2:2} = 5$$

With respect to our formal model for classifying OCR errors, the last remaining question is one of ambiguity. While the "quick brown fox" example may seem relatively straightforward, can we really be certain the 1:2 substitution `T` → `'l` was not caused by a speck of dirt being recognized as `''` (i.e., an apostrophe insertion), followed by a 1:1 substitution `T` → `l`? More severely damaged lines introduce even more uncertainty (see, for example, Appendix C). In addition, the trace-back table itself may contain multiple minimum-cost edit paths; in these cases we choose one representative path arbitrarily. At this early stage in our research, we are able only to note that such ambiguity exists, and that from our experience the vast majority of OCR errors seem to be correctly classified by our procedure.

3. SOURCE DATA – THE MOBY-DICK EXPERIMENT

We gathered statistics on errors made by a commercial OCR package when processing the Hendricks House edition of Herman Melville's novel *Moby-Dick*. The on-line version of the text, as prepared by Professor E. F. Irey, was obtained from the Guttenberg Project at The University of Illinois. We selected this particular source document for our experiments because it is well known and provides a significant body of text with no specific formatting requirements.

We felt it was desirable to work from a text for which we had an on-line, canonical representation. The alternative, using existing paper documents, requires a significant amount of data entry and verification after which a certain number of transcription errors are bound to remain. While our use of the Guttenberg Project text did not eliminate all inconsistencies with published editions of *Moby-Dick* (for example, the version we obtained was missing Chapter 74), it is still a freely available corpus which required no manual editing on our part. This should aid in the reproducibility of the results described in this paper.

Similarly, we thought it important to use an actual work of English literature as opposed to a random assortment of letters or words to create a reasonable context for the OCR software. It is well-known that recognizing random letters produces significantly different OCR results than real text.

Since our goal was to concentrate on character recognition errors and not format-related errors, we adopted the following simple rules:

(1) Lines and pages break in the same place in all test cases. As a result, lines consist of at most 79 characters (as determined by the widest font, Courier).

(3) Chapter headings start on a new line.

(4) The first paragraph after a chapter heading starts on a new line.

(5) Tabs and multiple spaces are replaced by a single space.

(6) Blank lines are removed.

These conventions undoubtedly minimized the formatting errors we saw during our tests. Still, we were not totally successful in preventing all such errors, as we shall discuss later.

The data preparation consisted of preprocessing the text using an *awk* script. The output was left-justified in a single column with an average line length of 76.4 characters. There are a total of 193,411 spaces in our *Moby-Dick*. The most frequently occurring non-space characters are "e" (113,484), "t" (84,050) and "a" (73,891). All the lower-case letters occur more than 10,000 times except for the letters "j" (815), "k" (7,736), "q" (1,202), "v" (8,313), "x" (1,186) and "z" (614). The frequency-of-occurrence for capitalized letters is in the hundreds. Fewer than 100 of each of the digits "0" through "9" occur in the text. The most frequently occurring punctuation mark is the comma (18,809) and the least frequent is the exclamation point (1,719). The character frequency for lower case letters is high enough for us to draw reliable conclusions based on this data, but the frequency of capital letters and digits is probably too low.

The preprocessed text was printed on a 400 dpi NeXT laserprinter in 10-point Times, Helvetica, and Courier fonts. Each page was examined to insure uniform print quality with no obvious extraneous marks. The majority of the pages in each version contain 48 lines of text, but several have 49 lines as a result of the pagination algorithm of the word processor we were using. On average, a page of text has 3,662 characters.

We scanned the printed pages using an HSD Scan-X Professional flat-bed scanner at a resolution of 300 dpi. The pages were fed into the scanner using the automatic document feeder (ADF) attachment in 10 chapter batches (ranging between 13 and 33 pages). While we uncovered a small number of document misfeeds in our later analysis, we observed no obvious paper-handling problems at the time of the scanning.

The HSD scanner software generates one-bit TIFF images which we input to the OCRServant v2.03 OCR package. The OCR software produces output in Rich-Text Format (RTF) which we converted to standard ASCII text using an RTF-to-ASCII filter.

By default, OCRServant crops all four margins tightly around the text. It adds leading spaces to a line only when necessitated by its interpretation of the document format (e.g., a spurious character in the left margin of the page will cause leading spaces to be added to the remaining lines). The post-processing RTF-to-ASCII filter did, however, add trailing spaces to the ends of some lines and blank lines to the ends of some pages. These filter-induced effects were removed from the OCR text. We used file difference software to verify that there were no other undesirable modifications to the data.

The OCR outputs for the three test cases were then compared to the original ASCII text using the algorithm described in Section 2.

To confirm certain of our results, we employed a second OCR Package, Calera v1.06. The Calera software differs significantly from OCRServant in its treatment of margins: it makes strong assumptions about character widths and nearly always inserts extra spaces before each line. As a result, overall error counts are not comparable in the two cases. The results reported in this paper are from OCRServant.

4. OCR ERROR CHARACTERISTICS

Simply comparing the original and OCR'ed version of the text reveals a set of common OCR misinterpretations. In Times, "r" is sometimes mistaken for "t" and "e" for "c". Such errors occur regardless of the position of the characters in the text. Furthermore, our experiment reveals that each font has its own characteristic error set. For example, in Helvetica it is far more likely that an "l" will be mistaken for an "I", or a comma for a period, than either of the previously mentioned Times errors.

A line-by-line comparison of the data also reveals a set of common errors involving multiple characters. In Times, "r" followed by "n" is frequently recognized as "m", whereas in Courier the pair "ow" can be mistaken for "OW". Our algorithm correctly classifies these errors.

Some more complex examples are shown in Appendix A. In the first, the bitmap of the word "gorgeous" is interpreted as "90T9eOUS". Apparently the OCR software's confusion of "g" and "9" forces subsequent errors such as "OUS" for "ous". This illustrates the context-sensitivity of OCR errors: it seems unlikely the OCR software would have made the second mistake if it had not made the first. Our current classification algorithm is unable to catch such dependencies: an analysis like this requires a significant amount of human reasoning and sometimes even "guesswork." Instead, our approach classifies each error separately, in isolation from the surrounding context.

The second example in Appendix A shows an OCR text line littered with space insertions, evidently caused by a mistaken assumption regarding inter-character spacings. Notice that, apart from the insertions, all of the characters were interpreted correctly. Again, we treat this as 13 distinct space insertions at present.

The last two examples illustrate text damaged by a document misfeed and by a "dropped" scan-line. These errors are representative of a class of mechanical failures caused by the scanning device and can only be detected by examining the bitmap.

An analysis of our OCR data indicates that the more catastrophic errors illustrated in Appendix A account for only a small percentage of the total errors. This no doubt reflects the care we used in preparing our source documents for OCR. Hence, we can be fairly confident that the results we are about to present are "correct" (relative to the constraints discussed in Section 3) in the sense that a human expert would agree with our classifications. To confirm this, we examined the first 100 pages of the Times version of *Moby-Dick* by hand. We found that approximately 3% of the errors were misclassified.

In the remainder of this section we report our findings for the *Moby-Dick* experiment. We examine cumulative and distinct error-growth (shown graphically in Appendix B). Finally, we discuss the composition of the OCR error sets, paying particular attention to the differences between the three fonts in question.

4.1. Basic measurements

Table 1 presents some basic statistics for the original text and the OCR'ed versions. The last row of the table lists the number of errors found using our classification algorithm. Typically there are more characters in the OCR text than in the original – in the case of Helvetica, this amounts to over one thousand additional characters. Most of this difference can be explained by a large number of space insertions. As we shall show later, space errors account for between 10% and 43% of all the OCR errors we saw. We also note that no text lines were deleted or inserted in the processing of any of the data sets. OCR of the Times version was more error prone than the other two fonts. Helvetica was interpreted the most accurately.

	Original	*Times*	*Helvetica*	*Courier*
Characters	1,179,194	1,180,122	1,180,259	1,179,970
Lines	15,235	15,235	15,235	15,235
Pages	322	322	322	322
Errors	0	4,996	2,304	2,905

Table 1. Basic statistics for the Moby-Dick experiment.

Overall, the OCR software did quite well on all three test cases. Table 2 shows that the per-character accuracy for Times, for example, is 99.6%, while the per-line accuracy is 67%. However, even at an accuracy as impressive as 99.6%, an OCR user should still expect to see 15 or 16 errors on an average page. Between 1 in 3 and 1 in 6 lines of text will contain an error, depending on the font. In our testing, nearly every full page incurred at least one OCR error (for Times, none of the pages was OCR'ed perfectly).

4.2. Error growth

The first graph in Appendix B shows total and distinct error growth for the three fonts. Each grouping of 500 lines along the X-axis corresponds to approximately 40,000 characters. The curves labeled "Total Errors" represent a count of all OCR error occurrences up to that point in the text. As expected, the errors for Times grow more rapidly than those for either Helvetica or Courier. The curves labeled "Distinct Errors" denote the growth of the characteristic error sets, the number of different errors seen for a given font. By the end of the data, the total error count is almost 10 times greater than the distinct error count.

Error Rate	*Times*	*Helvetica*	*Courier*
Per-Character	4.23×10^{-3}	1.95×10^{-3}	2.46×10^{-3}
Per-Line	3.28×10^{-1}	1.51×10^{-1}	1.91×10^{-1}
Per-Page	15.5	7.16	9.02

Table 2. OCR error rates for the three fonts.

The sharp jumps in total errors, for Times in the interval [1,500–2,000], for Helvetica in the intervals [3,000–3,500] and [12,500–13,000], and for Courier in the interval [5,500–6,000], are all the result of a spurious character in the left margin of the page causing four spaces to be inserted at the beginning of every line. Thus, approximately 160 of the space errors for Times and Courier and 320 of the space errors for Helvetica are the side-effects of other, seemingly unrelated, incidents. For Helvetica, this means roughly one third of all space errors are caused by two OCR failures. In the case of Courier, over half the space insertions are due to one anomalous event.

4.3. Composition of the error sets

The relative impact of space errors is more clearly demonstrated in the next three graphs we present in Appendix B. These figures show the cumulative growth of both space and non-space errors as well as the growth of distinct errors for Times, Helvetica, and Courier, respectively. If anything, the simple document formatting rules we followed should tend to minimize space errors. Clearly "white space is hard," as we observed earlier.

Table 3 shows a breakdown of deletion and insertion errors for non-space and space characters, as well as the total for all characters. Since the current classification code regards space errors as either deletions or insertions (spaces never participate in substitutions), all such errors are accounted for in the table. Note that space insertions are far more common than space deletions,

and that the latter are a major problem only in Times. Courier seems relatively immune to space errors; undoubtedly this is because it is a mono-spaced font.

	Times			*Helvetica*			*Courier*		
Error Type	*Non-Space*	*Space*	*Total*	*Non-Space*	*Space*	*Total*	*Non-Space*	*Space*	*Total*
Deletion	152	328	480	20	0	20	18	5	23
Insertion	110	1,650	1,760	67	999	1,066	199	293	492
All	3,018	1,978	4,996	1,305	999	2,304	2,607	298	2,905

Table 3. Breakdown of OCR non-space and space errors.

The debate as to whether space errors are "important" is one we choose to ignore at this point. In some applications space errors can be ignored, in others they may be disastrous.

The next three graphs in Appendix B illustrate the growth of the six OCR error-types we can classify using our current algorithm (single character deletions, insertions, and substitutions, and 1:2, 2:1, and 2:2 substitutions) for Times, Helvetica, and Courier, respectively. The graphs show the number of distinct errors of a given kind encountered as the text is processed sequentially. Although the curves seem to be leveling off (especially for Helvetica), it is not possible to say with certainty that a fixed-size characteristic error set exists for any of the fonts. We believe that more data must be examined to confirm or disprove this hypothesis.

A more definitive statement can be made regarding the compositions of the error sets. A visual comparison of the "Growth of Distinct Errors" graphs shows fundamental differences between the three fonts. For example, the four forms of substitution play equal roles in the case of Times, but 1:1 substitutions are clearly the dominating factor for Helvetica. Courier experiences almost no length-reducing errors (i.e., deletions or 2:1 substitutions).

To account for the intervals of accelerated error growth, we examined by hand all severely damaged lines of text. Gross physical defects in the bitmap can induce uncharacteristic (i.e., random-looking) errors – recall the examples of Appendix A. We modified our classification code to flag lines containing six or more OCR errors. These were run through a second OCR package to verify that the problem was indeed with the scanned TIFF image.

We found that the 14 worst Times lines generated a total of 104 OCR errors. There are typically one or two such lines for every 500 lines of "normal" text. Their concentration is much larger in the interval [4,800–6,000] where there are three damaged lines, and in the interval [12,000–14,000] where there are nine damaged lines. Close inspection of the associated graph shows a jump in the number of distinct errors in these intervals. On the other hand, the interval [6,700–9,800] contains no badly damaged lines and the corresponding region of the distinct error growth curve is relatively flat.

In the case of Helvetica, we determined that the six worst lines contributed 83 OCR errors to the total. There are two damaged lines in the interval [3,500–4,000], two more in the interval [8,000–8,500], and one particularly bad line in the interval [11,000–11,500]. As expected, the graph in question shows more rapid growth in these regions. In contrast, lines in the interval [4,000–4,500] suffered no unusual defects attributable to scanning failures, as evidenced by the flat slope of error growth curve.

Error Type	*Total*	*Distinct*	*Most Frequent Occurrences*
1:1 Substitution	1,335	159	r→t (134); e→c (86); d→~ (84); o→~ (70); d→Q (53); a→n (47)
Deletion	480	18	space (328); - (67); , (21); a (18); r (12); . (9)
Insertion	1,760	19	space (1,650); ~ (44); I (12); t (12)
1:2 Substitution	329	110	k→lr (43); a→~b (26); n→tl (13); n→ll (12)
2:1 Substitution	681	95	rn→m (341); fl→B (82); fl→a (20); ru→m (18); tl→k (17)
2:2 Substitution	411	150	ll→II (48); lk→kl (47); ad→~~ (31); tk→ l (26); rm→nn (25)
Total	4,996	551	

Table 4. OCR error classification results for Times.

Finally, we discovered that 14 lines in the Courier text were responsible for 162 of the OCR errors. In the interval [0–500] one specific line adds 17 errors, in [8,400–10,000] four lines were damaged, and in [12,300–13,600] there are four bad lines including one with 15 errors. The distinct error graph shows accelerated growth in the corresponding locations because of these events. The

interval [5,000–5,500] made it through the scanning process relatively unscathed and, as a result, few new errors appeared in this region.

Tables 4, 5, and 6 summarize our classification results for Times, Helvetica, and Courier. For each type of error we show the total occurrences, the number of those that are distinct, and a list of the most common errors.

The Times data (Table 4) is notable for the large number of multi-substitutions and line-length altering errors. Of course, the most common errors involve spaces – these account for almost 90% of the deletions and insertions we saw. Still, there are over 400 non-space errors that increase the length of a line, and over 800 non-space errors that decrease the length of a line. It is also intriguing to note the relatively high frequency of "rn → m" errors; an examination of the source text shows that the OCR software makes this error about 25% of the time.

Table 4 also illustrates the misclassification issue we raised earlier. The 2:2 substitution "ll → II" should probably be classified as two independent 1:1 substitutions "l → I" (even though this latter error is not one of the more common 1:1 substitutions for Times, as it is for Helvetica). We are currently examining ways of extending our algorithm to handle such cases.

Error Type	*Total*	*Distinct*	*Most Frequent Occurrences*
1:1 Substitution	881	114	,→. (95); ;→: (65); l→I (63); I→l (61); t→f (31)
Deletion	20	8	f (5); ' (4); - (4)
Insertion	1,066	11	space (999); n (21); ~ (15)
1:2 Substitution	57	41	k→I< (4); n→rl (4), d→cl (3)
2:1 Substitution	38	21	e-→space (4); k'→K (4); t'→~ (4); zz→u (4); kf→M (3)
2:2 Substitution	242	64	ll→II (58); rw→MI (51); II→11 (26); rw→nr (9); tw→nr (9)
Total	2,304	259	

Table 5. OCR error classification results for Helvetica.

Unlike Times, Helvetica (Table 5) experiences almost no line-length decreasing errors (i.e,. deletions and 2:1 substitutions). Other than space errors, 1:1 substitutions are by far the most prevalent type. Interestingly, Helvetica and Times share some points of commonality: in both cases the letter "k" is subject to mistakes in segmentation. At first, certain of the errors seemed entirely "random" to us – "rw → MI" is a good example – until we realized that the number of vertical strokes (in this case five) is usually preserved in such multi-substitutions. This is also probably indicative of segmentation failure.

Error Type	*Total*	*Distinct*	*Most Frequent Occurrences*
1:1 Substitution	1,896	196	t→~ (164); ,→. (157); o→~ (107); a→~ (67); e→c (54); t→+ (38)
Deletion	23	7	- (12); space (5)
Insertion	492	23	space (293); . (76); ~ (52); : (11)
1:2 Substitution	311	125	n→Il (44); g→cl (34); n→ll (14)
2:1 Substitution	4	4	
2:2 Substitution	179	99	ow→OW (26); ou→OU (19); oo→OO (6)
Total	2,905	454	

Table 6. OCR error classification results for Courier.

The Courier data, presented in Table 6, exhibits even fewer line-length reducing errors than Helvetica. Errors that increase the length of a line are common, however. "Capitalization" errors seem to be an anomaly peculiar to Courier (e.g., "ow → OW"), but this font shares the "e → c" and "n → ll" errors with Times. Finally, commas and periods are more of a problem here, despite the fact that they look pretty much the same across all three fonts. This is probably because forcing a constant character width places more white-space around these small punctuation marks.

5. CONCLUSIONS

In this paper we have reported the results of a large-scale experiment using a novel dynamic programming approach for the classification of OCR errors. We extended a well-known algorithm to handle multi-character substitutions of the kind that commonly arise in OCR, in addition to the more traditional substitutions, deletions, and insertions.

In a simply formatted text totaling 1.2 million characters, we identified between 2,300 and 5,000 OCR errors depending on the font. While a significant fraction of the error occurrences involve space deletions and insertions, we encountered a broad range of interesting error types, some intuitive, some difficult to explain. Even at accuracies greater than 99.6%, there is considerable structural complexity in the characteristic error set for a given font.

	Times		*Helvetica*		*Courier*	
	First	*Last*	*First*	*Last*	*First*	*Last*
Error Type	*1,000 Lines*	*1,000 Lines*	*1,000 Lines*	*1,000 Lines*	*1,000 Lines*	*1,000 Lines*
1:1 Substitution	43	7	16	0	47	4
Deletion	8	1	5	0	4	0
Insertion	6	1	1	0	8	2
1:2 Substitution	21	3	2	1	14	7
2:1 Substitution	14	1	5	0	0	0
2:2 Substitution	23	7	6	1	18	5
Total	115	20	35	2	91	18

Table 7. Appearance of new OCR errors at the beginning and end of Moby-Dick.

As might be expected, the number of distinct errors grows much more slowly than the number of total errors. However, our data does not allow us to state with certainty that the size of the error set for a particular font will eventually reach a constant. Table 7 shows the number of new errors encountered when processing the first 1,000 lines of *Moby-Dick* versus the last 1,000 lines. In the case of Helvetica, the font with the overall highest accuracy, only two previously unseen errors appear towards the end of the document. This suggests that more data must be examined to confirm or disprove our hypothesis.

Visual inspection of the bitmaps for lines severely damaged during scanning allowed us to explain periods of accelerated error growth. It may be that certain error-types are not characteristic of a particular font or OCR package, but are instead induced by physical failures in the image acquisition process.

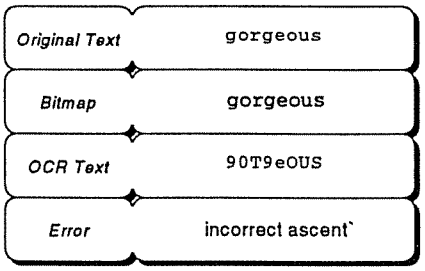
Error Type	*Errors in Common*	*Total Errors*	*Overlap*
1:1 Substitution	41	318	12.9%
Deletion	6	18	33.3%
Insertion	7	30	23.3%
1:2 Substitution	5	240	2.1%
2:1 Substitution	0	118	0.0%
2:2 Substitution	2	302	0.7%
Total	61	1,026	5.9%

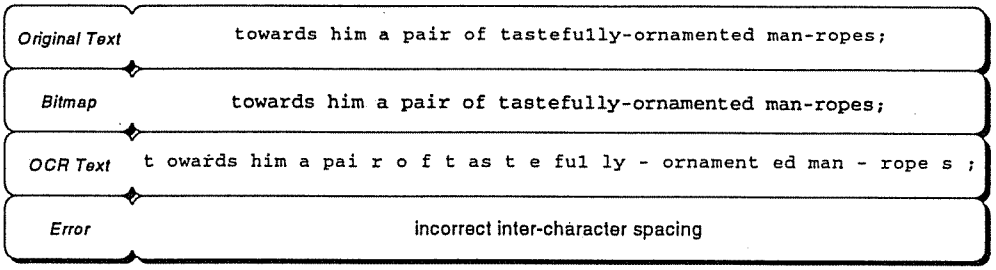
Table 8. Overlap of OCR error sets for the three fonts.

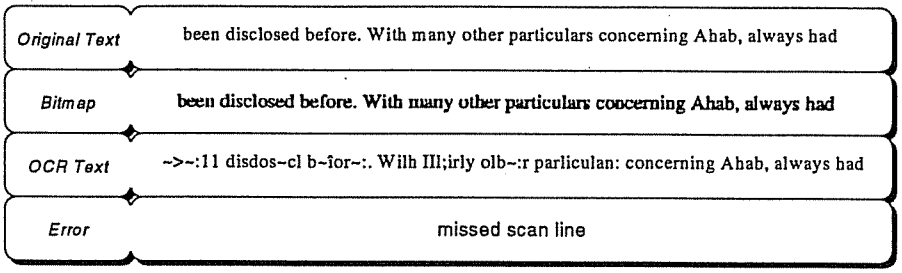
Our data shows that the composition of OCR error sets varies significantly from font to font. Obvious differences include the total number of deletions and insertions, the number of length-altering errors, and the relationship between the three forms of multi-substitutions. The identity of the errors themselves is also a distinguishing factor. Table 8 illustrates this point; the three fonts are seen to share many of the simpler types of errors, but more complex errors are nearly always unique to a specific font.

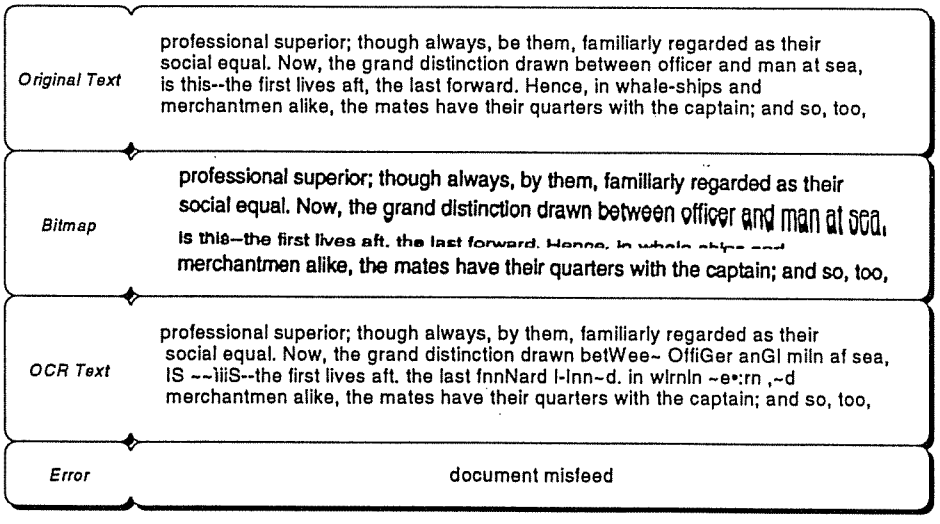
In addition to the obvious scanner-induced errors illustrated in Appendix A, we also observed a more subtle periodic error source introduced by the use of an automatic document (i.e., sheet) feeder. The last graph in Appendix B plots the number of errors versus the line on the page for Courier font. In examining this figure, sharp spikes in the error rate on nearly every third or fourth line are immediately evident. This period corresponds to the observed starting and stopping of the document feeder as each page is scanned (and, presumably, the scanner's internal memory buffer is uploaded to the host computer). Similar results were obtained for the other fonts, and are examined in more detail in a later paper.[8]

6. APPENDIX A – OCR ERROR EXAMPLES

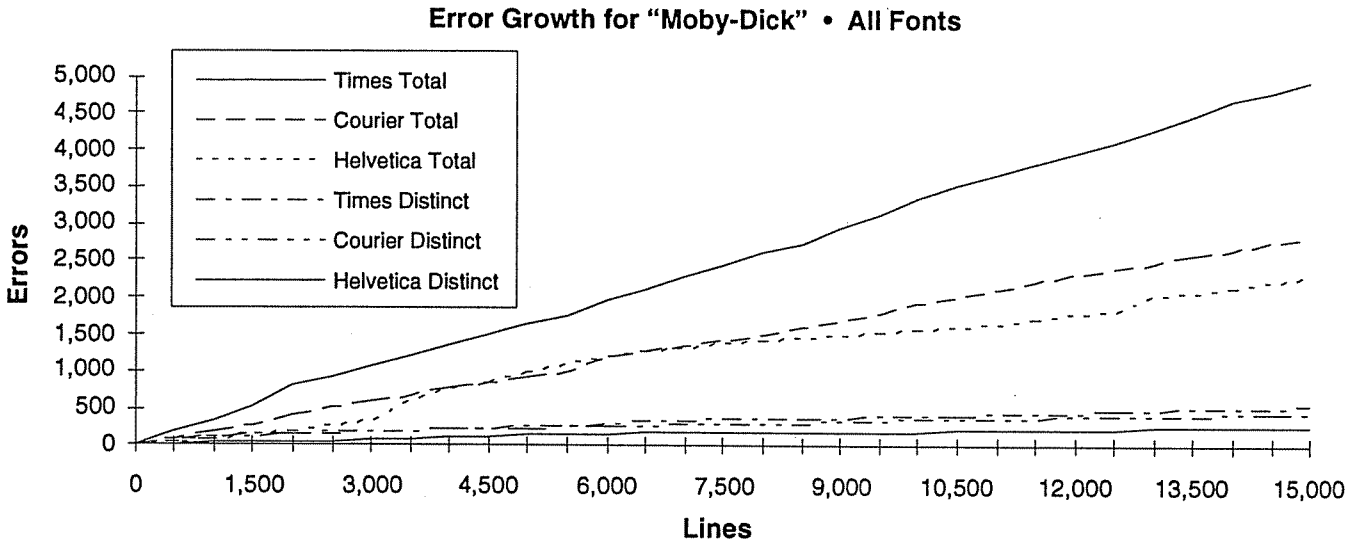


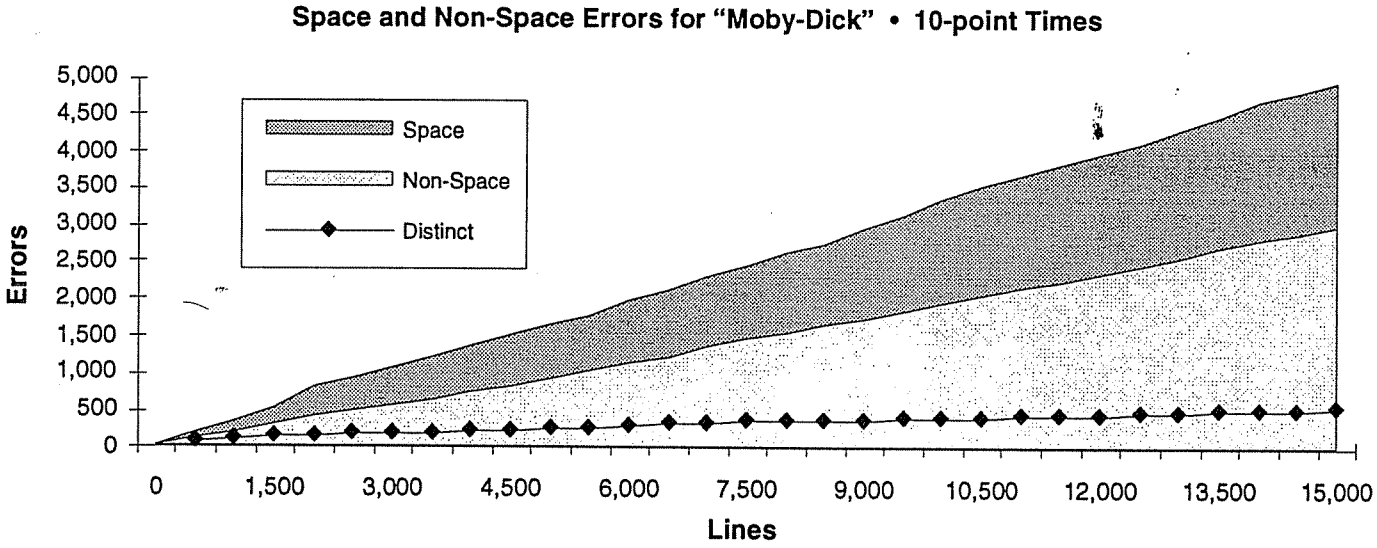


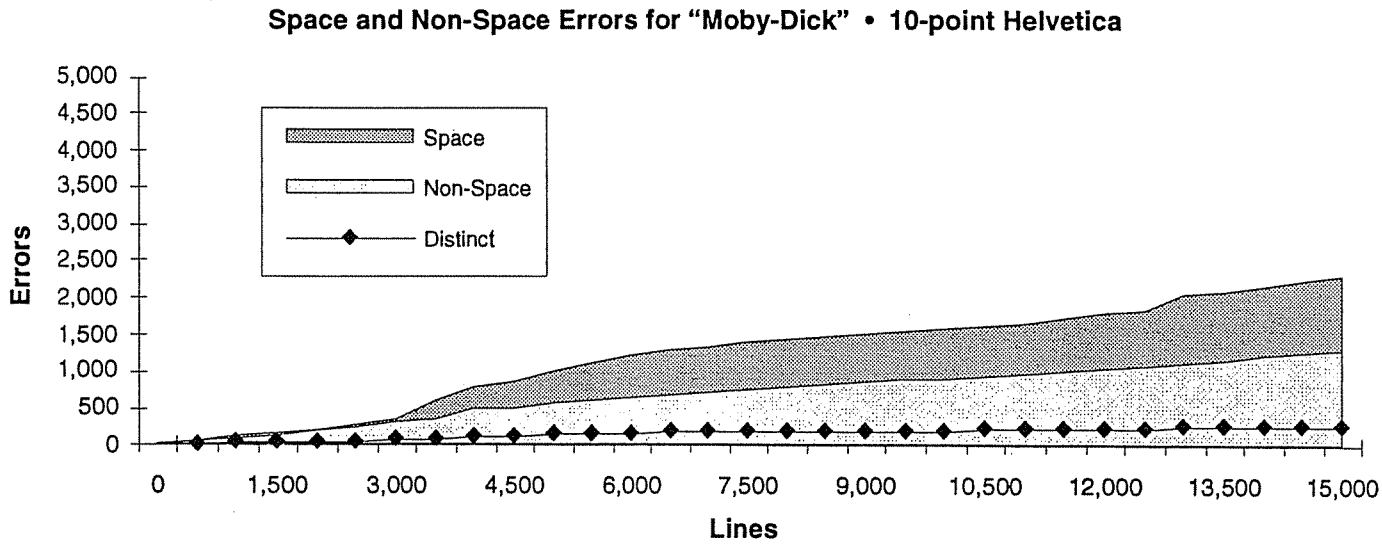


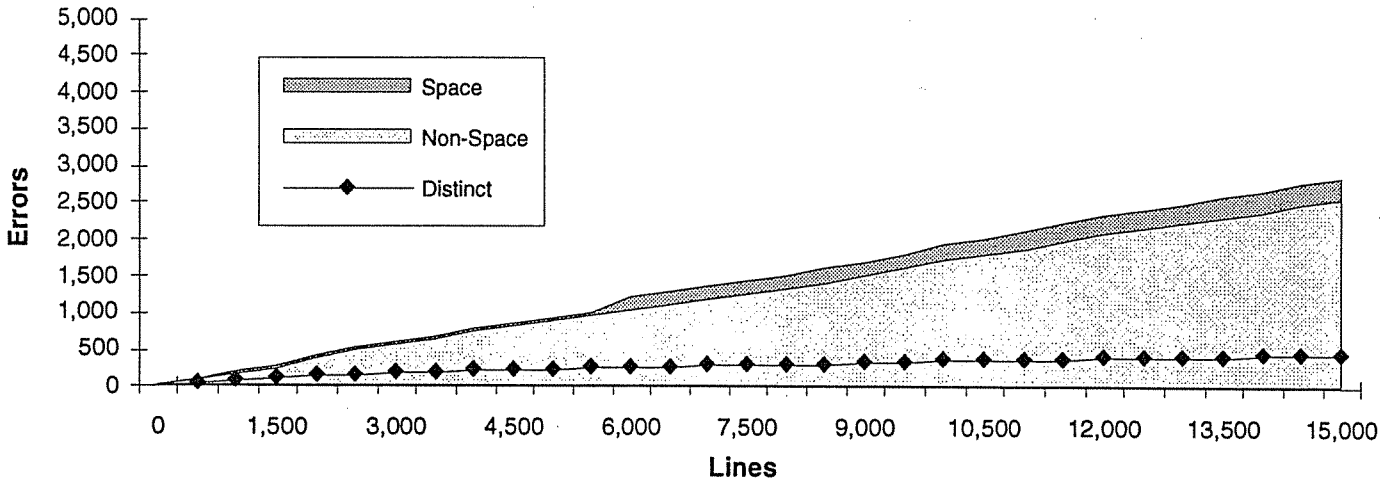


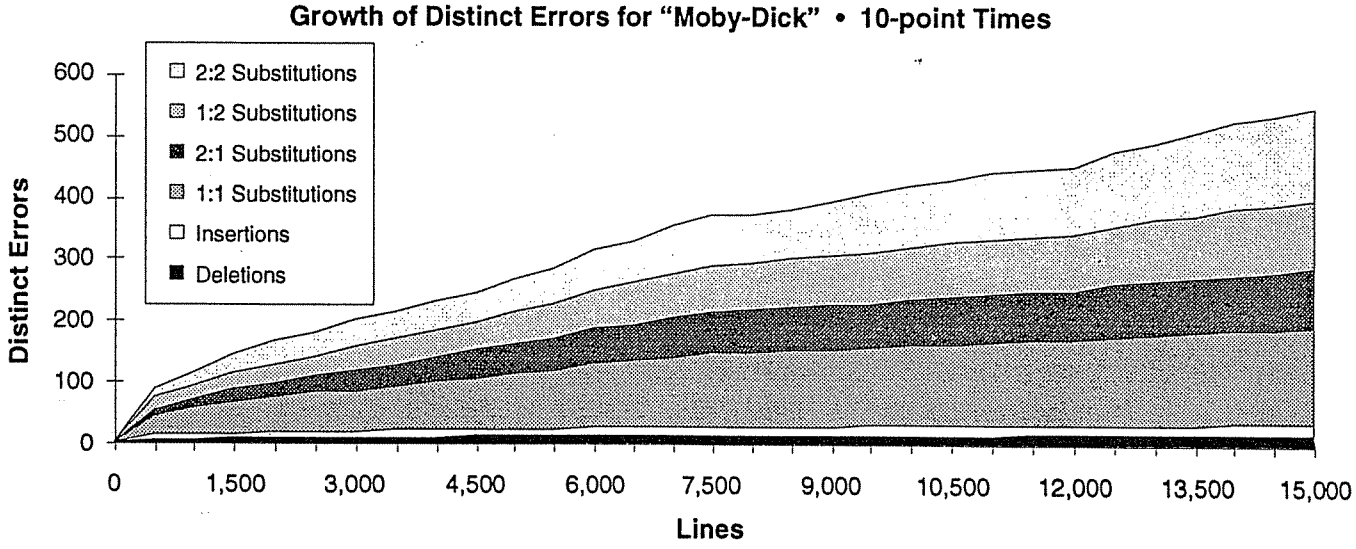
7. APPENDIX B – OCR ERROR ANALYSES

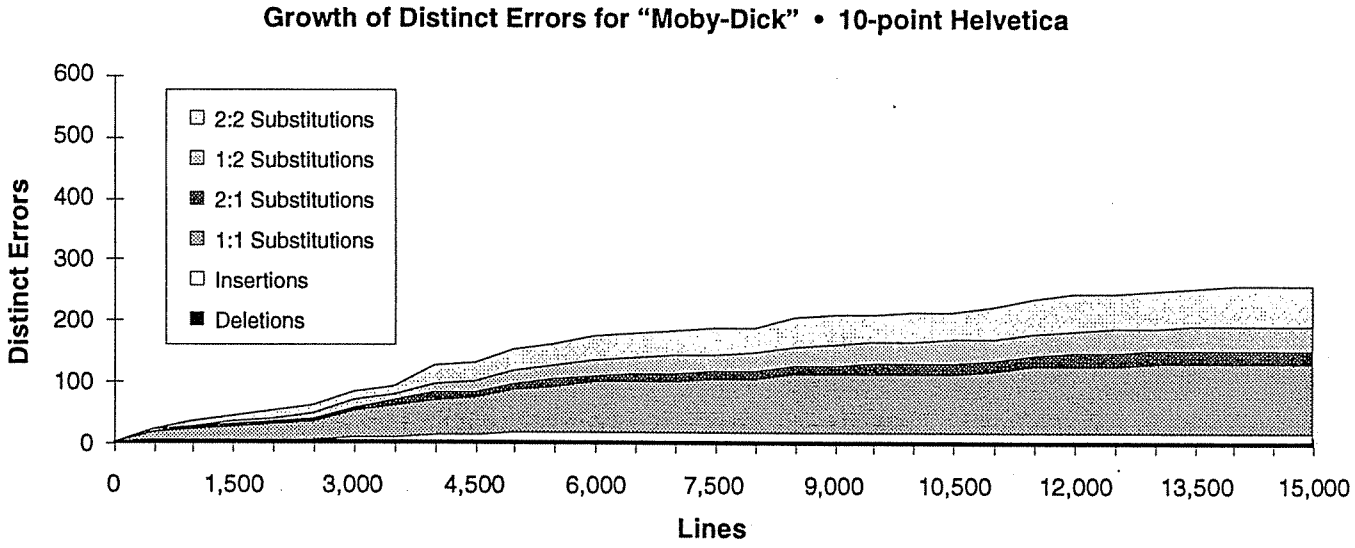


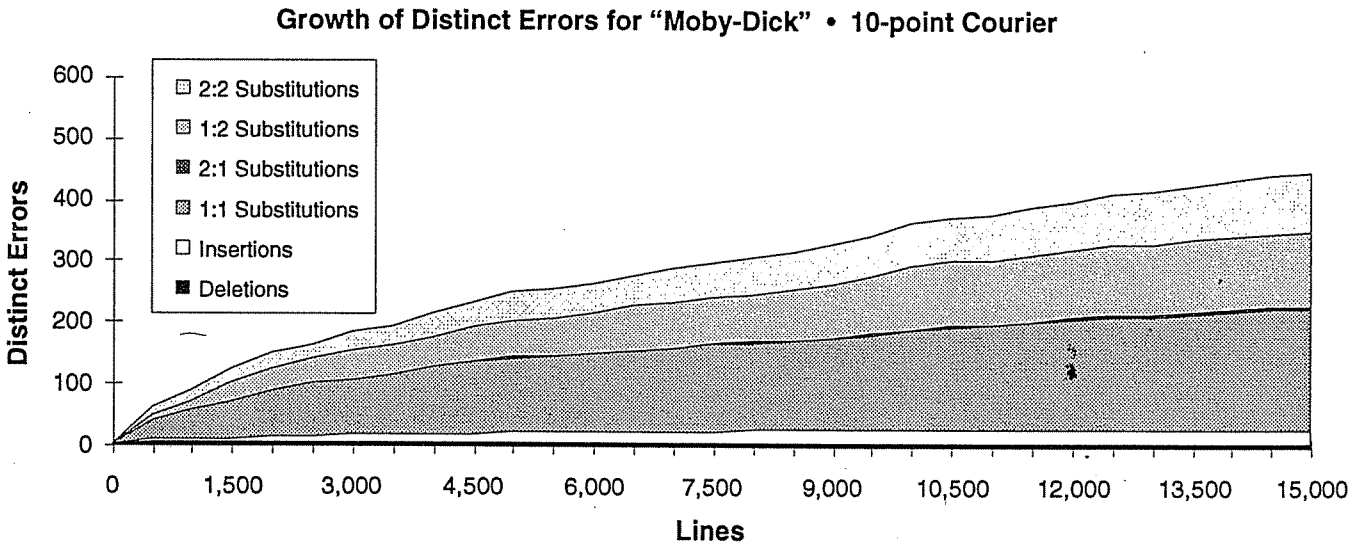


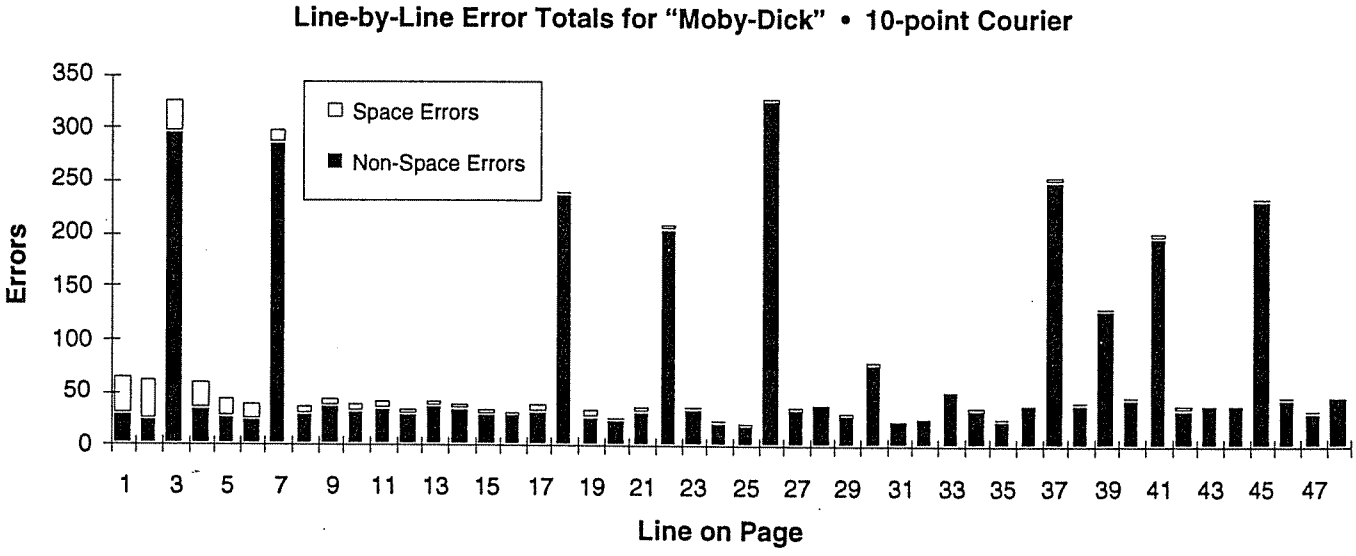

Space and Non-Space Errors for "Moby-Dick" • 10-point Helvetica
Errors
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Space
Non-Space
Distinct
0
1,500
3,000
4,500
6,000
7,500
9,000
10,500
12,000
13,500
15,000
Lines


Space and Non-Space Errors for "Moby-Dick" • 10-point Courier
Errors
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Space
Non-Space
Distinct
0
1,500
3,000
4,500
6,000
7,500
9,000
10,500
12,000
13,500
15,000
Lines


Growth of Distinct Errors for "Moby-Dick" • 10-point Times
Distinct Errors
600
500
400
300
200
100
0
2:2 Substitutions
1:2 Substitutions
2:1 Substitutions
1:1 Substitutions
Insertions
Deletions
0
1,500
3,000
4,500
6,000
7,500
9,000
10,500
12,000
13,500
15,000
Lines


Growth of Distinct Errors for "Moby-Dick" • 10-point Helvetica
Distinct Errors
600
500
400
300
200
100
0
2:2 Substitutions
1:2 Substitutions
2:1 Substitutions
1:1 Substitutions
Insertions
Deletions
0
1,500
3,000
4,500
6,000
7,500
9,000
10,500
12,000
13,500
15,000
Lines


Growth of Distinct Errors for "Moby-Dick" • 10-point Courier
Distinct Errors
600
500
400
300
200
100
0
2:2 Substitutions
1:2 Substitutions
2:1 Substitutions
1:1 Substitutions
Insertions
Deletions
0
1,500
3,000
4,500
6,000
7,500
9,000
10,500
12,000
13,500
15,000
Lines


Line-by-Line Error Totals for "Moby-Dick" • 10-point Courier
Errors
350
300
250
200
150
100
50
0
Space Errors
Non-Space Errors
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
Line on Page

8. REFERENCES

1. S. Kahan, T. Pavlidis, and H. S. Baird, "On the Recognition of Printed Characters of Any Font and Size," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, no. 2, March 1987.

2. J. Schürmann, et. al., "Document Analysis – From Pixels to Contents," *Proceedings of the IEEE*, vol. 80, no. 7, July 1992.

3. D. Lopresti and J. Sandberg, "Certifiable Optical Character Recognition," *Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR 93)*, October 1993, pp. 432-435.

4. S. V. Rice, J. Kanai, and T. Nartker, "An Evaluation of OCR Accuracy," 1993 Annual Report, UNLV Information Science Research Institute, pp. 9-33.

5. R. Bradford and T. A. Nartker, "Error Correlation in Contemporary OCR Systems," *Proceedings of the First International Conference on Document Analysis and Recognition (ICDAR 91)*, October 1991, pp. 516-524.

6. D. Sankoff and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, 1983.

7. K. Taghva, J. Borsack, B. Bullard, and A. Condit, "Post-Editing through Approximation and Global Correction," 1993 Annual Report, UNLV Information Science Research Institute, pp. 57-68.

8. J. Esakov, D. Lopresti, J. Sandberg, and J. Zhou, "Issues in Automatic OCR Error Classification," to be presented at the *Third Annual Symposium on Document Analysis and Information Retrieval*, April 1994.