Validation of Image Defect Models for Optical Character Recognition

*Yanhong Li** *Daniel Lopresti†* *George Nagy‡* *Andrew Tomkins§*

June 5, 1995

Abstract

In this paper, we consider the problem of evaluating character image generators that model distortions encountered in optical character recognition (OCR). While a number of such defect models have been proposed, the contention that they produce the desired result is typically argued in an ad hoc and informal way. We introduce a rigorous and more pragmatic definition of when a model is accurate: we say a defect model is validated if the OCR errors induced by the model are indistinguishable from the errors encountered when using real scanned documents. We describe four measures to quantify this similarity, and compare and contrast them using over ten million scanned and synthesized characters in three fonts. The measures differentiate effectively between different fonts and different scans of the same font regardless of the underlying text.

Index Terms: *optical character recognition*, *document image defect models*, *OCR error classification*, *defect model validation.*

1 Introduction

Differences between documents account for far more of the variation in OCR error rates than do differences between classification methods adopted by the various OCR manufacturers [1]. Surprisingly, the error rate achieved on a given document by mature OCR systems varies at most by a factor of two, while the error rate between

*GARI Software, 293 Eisenhower Parkway, Suite 250, Livingston, NJ 07039.

†Matsushita Information Technology Laboratory, Two Research Way, Princeton, NJ 08540.

‡Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180.

§Computer Science Department, Carnegie-Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213.

documents within a given application may vary by as much as 100:1 (*e.g.*, from 90% to 99.9% accuracy). The quality of a document, from an OCR perspective, is therefore defined in practice by the error rate it induces.

OCR accuracy depends on document composition (typeface, point size, spacing); printing (ink-spread, strike-through, paper defects); copying (skew, streaking, shading); and digitization (blurring, sampling, thresholding). Other document manipulations, such as folding, microfilming, and facsimile transmission, may add further degradation. (In handprinted character recognition, the motivation of the writer is often the dominant factor.) Not even the digitization process is under the complete control of the OCR manufacturer – the prevalence of image databases, desk-top scanners, and digital facsimile requires accepting bitmaps generated by arbitrary, unknown scanning systems. A small fraction of highly degraded pages can significantly increase the overall error rate and the resulting post-editing cost [2].

In character recognition, it has proved difficult to model mathematically either the signal or the noise. Printed and handprinted characters simply are not amenable to concise formal description, and the sources of noise and distortion are manifold and complex. However, just as hope for the predictive modeling of the classification process has faded, there has been a marked resurgence of interest in pseudo-random defect models for generating large synthetic sample data sets. Although some aspects of these models are based on observable physical phenomena, compelling arguments can also be made for purely empirical or descriptive models [3].

The use of randomly-generated characters in place of real data was popular twenty years ago for the same reason it is popular today: it is much easier to generate large data sets from a few prototypes under program control than it is to scan, segment, and label real data. The earliest defect models for generating synthetic data for OCR were based on salt-and-pepper noise. The noise source produced independent and identically distributed (i.i.d.) random variables. Some researchers were amazed at how well even simple classifiers could cope with impressively large amounts of such noise.

Today's models are far more realistic at simulating the distortions that real OCR devices must face. They usually comprise a combination of deterministic and randomized sources of distortion that are parameterized to reflect the prevalence of various types of noise. Typically, typeface variations are modeled using prototypes from digital fonts. Degradations that arise from the imaging, copying, or digitization (scanning) processes are also treated. Most of the models described to date produce individual character samples rather than entire documents, but page-composition software can be used in combination with noise models to produce realistic renditions of entire pages.

While a number of defect models have been proposed in the literature, their proponents usually argue that they produce the desired result in an ad hoc and informal way. In this paper, we introduce a rigorous and more pragmatic definition of when a model is accurate: we say a defect model is validated if the OCR errors induced by the model are indistinguishable from the errors encountered when using real scanned documents.[1]

After providing a brief overview of defect models and their applications, we present our proposal for defect model validation, which is based on the comparison of OCR error sets. We describe four measures to quantify this similarity, and evaluate them using over ten million scanned and synthesized characters in three fonts. The measures differentiate effectively between different fonts and different scans of the same font regardless of the underlying text. We conclude the paper with a discussion of the effectiveness of this approach and areas for possible improvement.

2 Defect Models and Their Applications

Before presenting our method in detail, we describe several defect models, list some of their applications, and consider alternative approaches to validation.

salt-and-pepper The most common method of simulating noise in gray-scale images is to add normally distributed white noise. Another popular noise source is the multiplicative Poisson process. For bi-level images, however, it is simplest to randomly and independently switch black pixels to white with probability p, and white pixels to black with probability q.

clumps An improvement over this white-noise model, local correlation between pixels, was introduced by Suen and Wang in 1983 [4].

deterministic degradation Pavlidis generated isolated character samples from phototypesetter font descriptions of nine different typefaces using a scan-conversion algorithm. He used the model to show the effects on classification of horizontal and vertical scaling, rotation, sampling rate, and amplitude quantization threshold [5].

pseudo-random defect models Baird classified over one million character arrays generated from one hundred different digital fonts [6]. Some of the parameters are fixed, and some are probabilistic [7]. The variations modeled in an eight-million character test set that Baird contributed to a public-domain database [8] are: nominal point size, spatial sampling rate, character skew (rotation), horizontal and vertical scaling, horizontal and vertical translation, individual pixel displacement, Gaussian point-spread function, and threshold.

[1] In truth, the problem we address is more properly termed "defect model *invalidation*," since our null hypothesis is that the model and real data come from the same population. Proving that a particular model is always accurate seems inherently more difficult than showing that one is invalid. Still, we prefer the "positive" terminology.

digital bitmaps Experiments on bitmaps obtained from digital fonts remain popular. Recently, Jenkins and Kanai showed that commercial classifiers are often more accurate on clean scanned versions of the characters than on the digital prototypes themselves [9].

edge noise For their word recognition experiments, Khoubyari and Hull generated entire passages, in different typefaces, from the Brown Corpus [10]. They thickened the character strokes to produce some touching characters, then randomly erased some of the black pixels to simulate broken characters. By averaging several instances of the same word, they obtained impressive word-recognition results.

pixel morphology A pseudo-random defect model based on mathematical morphology has been developed by Haralick and his colleagues. In this model, the probability of a change in a pixel's value depends on its distance from the edge of the character [11].

perspective distortion Haralick has also modeled the geometric and photometric distortions introduced by the curl of the pages when copying bound volumes [11].

page distortion Buchman's page distortion model allows for varying the amount of rotation, blurring, line addition and drop-out, speckle, contrast, bleed-through, and amplitude quantization [12].

A trained observer would not confuse any of the published ensembles of artificial characters with scanned copy, but some individual characters do look authentic. One possible reason for the lack of realism is that few if any of these models simulate the random phase angle of spatial sampling, which gives rise to highly correlated noise [13, 14]. Yet the displacement of the character pattern relative to the scanning array is one of the most significant sources of distortion for cleanly printed characters in common point sizes. It can, for instance, change a sans-serif lower-case 'el' (*i.e.*, l) from a column two pixels wide to a column three pixels wide that may be mistaken for an upper-case 'eye' (*i.e.*, I). Furthermore, it is the only source of noise that affects uniformly all bi-level scanners with a given spatial resolution and that does not require any calibration.

The models mentioned above are intended for printed or typewritten characters only. However, models for handprinting have also been developed [15], and Plamondon and his colleagues have demonstrated a physiologically-plausible generative model for cursive writing [16]. As mentioned, individual variations in handprinting and writing often dominate noise due to the copying or scanning processes.

A realistic defect generator for digitized characters would have many applications. In the design phase, an accurate model could be used for training set augmentation and parameter optimization. From an analytic standpoint, it would facilitate controlled experiments, performance prediction, and sensitivity analysis. To the best of our knowledge, however, none of these applications has been successfully validated in the sense that the phenomena observed for synthetic data have been shown to transfer quantitatively to real data.

3 The Validation Problem

A number of approaches have been suggested for validation (or related) problems:

Turing test Can a human observer determine whether a given sample was synthetically generated? This test is not necessarily conclusive, since most humans have little experience judging digitized text.

learning Does the error rate on real data improve after a classifier has been trained on the simulated data? While this criterion may appeal to an OCR engineer, it does not allow distinguishing between two simulated data sets that improve the error rate equally, and is as much a function of sample size as model accuracy.

reproduction of observed samples One proposed measure of a defect model is its ability to reproduce pixel configurations observed in "live" text. At normal scan resolutions, exact reproduction is improbable; hence a matching or distance criterion must be introduced. Devising such a metric amounts to designing an *ad hoc* classifier.

hypothesis test on pixel distributions In principle, given the multinomial nature of the class-conditional pixel distributions, it is possible to test whether the distribution on the synthetic data is similar to the distribution on real data, as proposed by Kanungo and his colleagues [17]. However, it seems unlikely that a single distance measure could capture all possible differences between two pixel distributions. Constructing some type of stochastic metric to determine the statistical similarity of the two sample sets leads us to the next suggestion.

hypothesis test on feature distributions If the number of features is relatively small, and the distributions take on few values, then in principle this is a feasible approach. However, even for only 10 classes and 10-dimensional normally-distributed features, over 500 parameters must be estimated. Assuming that the features (or pixels) are class-conditionally uncorrelated just begs the question. A recent study compares two (real) hand-printed data sets using Karhunen-Loeve expansions of locally-averaged pixels [18]. But it is difficult to make a sound argument for the relevance of this particular set of features for *classification* (as opposed to mean-square reconstruction of the patterns).

cross-validation A measure of the compatibility of two data sets A and B, from the viewpoint of classification, is the relative error rate in the following experiment: a classifier is trained on a subset of A and tested on a *different* subset of A, and also on B. Then the converse experiment is performed, with training on a subset of B. If all four error rates are essentially the same, then there is reason to believe that A and B are similar, or at least equally easy (or difficult) for the given classifier [18]. This approach is most similar in spirit to our proposal.

calibration If the printing, copying, and scanning mechanisms are available to the experimenter, then physical model parameters could be measured using test patterns. Baird has demonstrated a method that recovers most of the parameters [19]. He has yet to calibrate the model against scanned data, though. Even if a set of parameters were obtained, the question of how completely the selected defects model real data would remain.

None of the above approaches appears to provide a satisfactory method of determining whether a given model represents those aspects of reality that are being investigated, *i.e.*, the defects that give rise to the misclassification of scanned data.

4 A Proposal for Validation

Our proposal is based on the widely accepted notion that the relationship between two entities may be quantified by means of a statistical comparison of *relevant* measurements taken on the entities. In our case, the two entities are (1) a real data set, and (2) a synthetic data set generated by the model to be validated. The novel part of our proposal is that the measurements consist of error distributions generated for each data set by a classifier of interest.

The general framework we envision for document defect model validation is presented in Figure 1. As depicted in the figure, we start from an on-line reference text. In the case of the control, shown on the left, the text is printed and scanned, thus introducing noise. On the right side, electronic text is transformed into a page image without ever having been rendered on paper. The defect model under study is then applied to mimic real-world damage.

When the two sets of document images are OCR'ed, they exhibit their own unique error characteristics. The similarity between these error distributions is calculated by the validation procedure. If the value returned indicates a close enough match, the defect model is considered validated, otherwise it fails. In this paper, we discuss four measures which can be used to perform the comparison. To gain some intuition about their performance, we present experimental data drawn from the printing, scanning, and OCR'ing of a large body of text using three different typefaces. The resulting analysis suggests, surprisingly, that this method may be powerful enough to obviate the requirement that the two error sets be generated from the same canonical text.

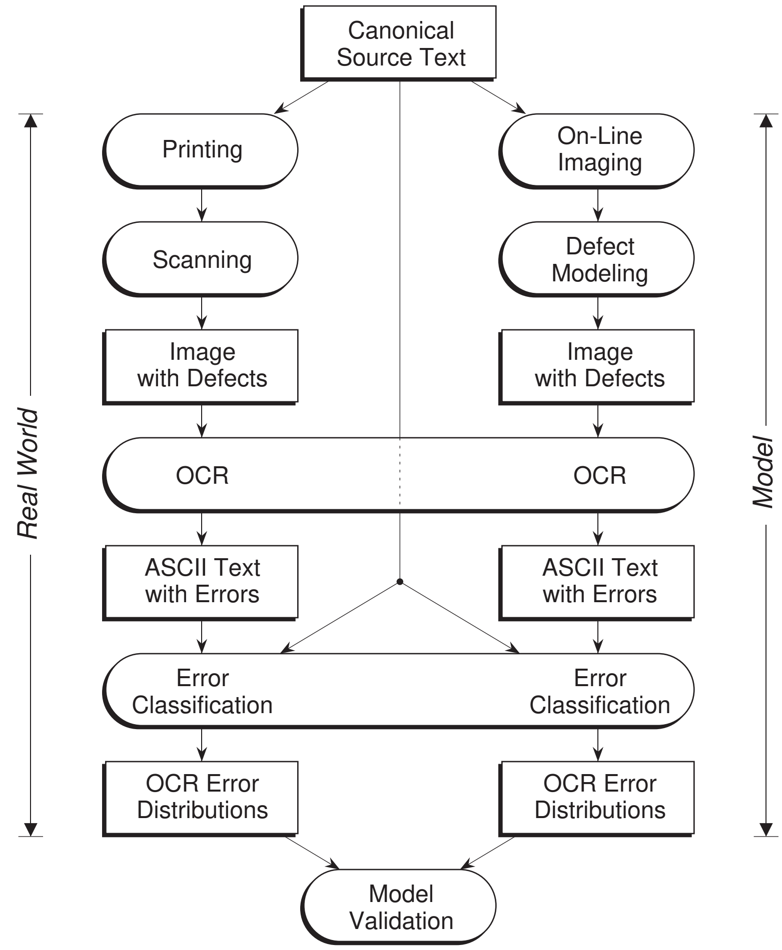

Figure 1: The document defect model validation process.

First, however, we must define the notion of an error distribution more formally. The OCR process introduces errors which can be identified and categorized using an error classification procedure. An OCR'ed data set induces a distribution over the set of errors based on the relative frequencies of each type.

For instance, if standard string edit distance is employed [20], the possible errors are single character deletions, insertions, and substitutions. This simple model does not, however, capture the notion of segmentation failures correctly. Consider, for example, the pattern $\{\texttt{m} \rightarrow \texttt{rn}\}$. Clearly this error represents a single event, a 1:2 substitution, rather than an insertion followed by a substitution (or vice versa). It is important that our classification procedure categorize such errors properly for two reasons. First, errors are infrequent in high-quality OCR; we wish to avoid introducing unnecessary variance. Second, analysis of the distributions is made tractable by assuming that errors are independent; highly conditioned patterns such as $\{\rightarrow \texttt{r}\}$

	Deletion	*Insertion*	*1:1 Sub*	*1:2 Sub*	*2:1 Sub*	*2:2 Sub*
Original Text	her,	were in	bar	and	flourish	forward
OCR Output	her	were in	bat	ancl	Bourish	foMIard
Error Pattern	,→	→ ⟨sp⟩	r→t	d→cl	fl→B	rw→MI

Table 1: Some observed OCR error patterns.

always immediately preceding {m→n} weaken the independence assumption.

The error classification procedure we use, as described in [21], handles arbitrary $g{:}h$ multi-character substitutions. In all of our tests, we set $0 \leq g, h \leq 4$, so patterns such as {nn→IlIl}, an error we observed in practice, are correctly identified as a single event. Generally, we only consider errors involving "printing" characters in our analyses (*i.e.*, we ignore "white-space" errors). This issue is discussed further in a later section.

The algorithm is based on a modified version of the string edit distance computation. If $S = s_1 s_2 \ldots s_m$ is the original (source) line, $R = r_1 r_2 \ldots r_n$ is the OCR (recognized) line, $c_{sub_{g:h}}$ is the cost of performing a $g{:}h$ substitution, and $d_{i,j}$ is the edit distance between $s_1 s_2 \ldots s_i$ and $r_1 r_2 \ldots r_j$, then the primary recurrence is

$$d_{i,j} = \min_{0 \leq g,h \leq 4} [d_{i-g,j-h} + c_{sub_{g:h}}(s_i \ldots s_{i+g-1}, t_j \ldots t_{j+h-1})] \tag{1}$$

Saving the optimal decisions as $d_{i,j}$ is computed makes it possible to enumerate the OCR errors after the edit distance phase of the algorithm completes. Table 1 shows some examples from our experiments.

More formally, let $E = \{e_1, e_2, \ldots, e_n\}$ represent the set of all possible error patterns. The similarity between the real and synthetic data sets is quantified by means of a real-valued probability distance function:

$$\mathcal{F} : \{\{p_1, p_2, \ldots, p_n\}, \{q_1, q_2, \ldots, q_n\}\} \rightarrow \mathcal{R} \tag{2}$$

where

$$\begin{aligned} p_i &= \text{Prob[pattern } e_i \text{ occurs in the real data]} \\ q_j &= \text{Prob[pattern } e_j \text{ occurs in the synthetic data]} \end{aligned}$$

The function $\mathcal{F}$ may be any one of a number of measures available for comparing two probability distributions, *e.g.*, Chernoff, Bhattacharyya, Matusita, Divergence, Patrick-Fisher, Lissack-Fu, Kolmogorov [22]. The discrete probability functions p_i and q_j are the error distributions of the real and synthetic data sets. Therefore,

$$\sum_{i=1}^{n} p_i = \sum_{j=1}^{n} q_j = 1$$

We now discuss two of the traditional measures in more detail, and describe two others we have developed for comparing OCR error distributions. The experimental results to be presented in Section 5 will be expressed in terms of these four measures, and we shall analyze them in greater detail at that time.

4.1 Traditional Measures

Two traditional measures for comparing probability distributions are the Bhattacharyya coefficient and Matusita distance. The former is defined by

$$\mathcal{B}(p,q) \equiv -\ln \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{3}$$

In general, the Bhattacharyya coefficient ranges from a value 0 for identical distributions to ∞ for completely different distributions.

Matusita distance is defined as

$$\mathcal{M}(p,q) \equiv \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{4}$$

The range for this measure is from 0 for identical distributions to $\sqrt{n}$ for completely different distributions.

4.2 The Vector Space Measure

The Vector Space model is widely used in information retrieval to quantify the similarity between two documents, or a query and a document [23]. In this approach, a text is represented by a vector of index terms or keywords. If a particular element is present in the document, the corresponding entry in the vector is set to 1; otherwise, it is set to 0. The similarity between two texts is then calculated as the inner-product of their term vectors. This measure is equal to the cosine of the angle between the two vectors, and hence is sometimes called "cosine similarity."

The Vector Space model has also been used recently in conjunction with OCR in a system that classifies documents based on keyword frequency using weighted cosine similarity [24].

We treat the two error distributions p and q as vectors by taking $\vec{V}^{(p)} = \langle v_1^{(p)}, \ldots, v_n^{(p)} \rangle$, where $v_i^{(p)} = p_i$ (and similarly for $\vec{V}^{(q)}$). The cosine similarity is then calculated as

$$\mathcal{V}(\vec{V}^{(p)}, \vec{V}^{(q)}) \equiv \frac{\vec{V}^{(p)} \cdot \vec{V}^{(q)}}{|\vec{V}^{(p)}||\vec{V}^{(q)}|} \tag{5}$$

where

$$\vec{V}^{(p)} \cdot \vec{V}^{(q)} = \sum_{i=1}^{n} \left(v_i^{(p)} \cdot v_i^{(q)} \right) \qquad \text{and} \qquad |\vec{V}| = \sqrt{\sum_{i=1}^{n} (v_i)^2}$$

The Vector Space measure ranges from 1 for identical distributions (*i.e.*, the angle between the vectors is 0°), to 0 for completely different distributions (*i.e.*, the angle is 90°).

4.3 The Coin Bias Measure

Unlike the previous measures, the Coin Bias measure is fundamentally new. The intuition behind it is as follows: Suppose an observer has full information about the distributions p and q. Now say we secretly choose one of p and q uniformly at random, generate OCR error patterns accordingly, and show them to the observer. The observer's "assignment" is to guess which distribution is being used to generate the data. For example, suppose p is highly likely to produce the error $\{\texttt{m} \rightarrow \texttt{rn}\}$, but q almost never does. An observer shown this error would be justified in guessing that p was the secret distribution.

For any particular error, the observer knows which distribution is more likely to produce it and can guess appropriately. Therefore, we judge that two distributions are similar if the observer can do little better than random at telling them apart. If they are identical, for instance, the observer will always be presented with an error that is equiprobable under both scenarios. Since a random guesser will be right with probability 0.5, a "similarity" value of 0.5 represents perfect validation of a defect model.

We begin by making the reasonable assumption that OCR errors are independent. Taking advantage of this, we can simplify the problem to the following question: when shown *one* error, chosen at random, what is the probability the observer guesses correctly?

An error of type e_i will be shown with probability $1/2(p_i+q_i)$. The most reasonable strategy for the observer is to guess p if $p_i > q_i$, and q otherwise; this leads to the minimum expected error rate (*i.e.*, the Bayes Risk) for what is effectively a two-class recognition problem. In this case, the observer will be right with probability $\max\{p_i, q_i\}/(p_i + q_i)$. Hence, the total probability the observer is right is

$$\begin{aligned} \mathcal{C}(p, q) &\equiv \sum_{i=1}^{n} \text{Prob[correct guess for } e_i] \qquad (6) \\ &= \sum_{i=1}^{n} \left(\frac{1}{2}(p_i + q_i) \frac{\max\{p_i, q_i\}}{p_i + q_i} \right) \\ &= \frac{1}{2} \sum_{i=1}^{n} (\max\{p_i, q_i\}) \end{aligned}$$

The Coin Bias measure is Bayesian, so we have implicitly assumed that the error patterns represent scanned images with probability 0.5, and synthetic images with probability 0.5. If these "prior probabilities" are different, the observer's strategy can be changed in an obvious way to reflect this.

5 Evaluation of the Validation Measures

In this section, we compare several large sets of real-world data with three goals in mind: to convey an intuitive feel for the validation process, to compare the four distance measures just described, and to examine the statistical nature of typical OCR error distributions. For the following, we used different fonts and text sources; intuitively, an effective validation measure should capture font-specific OCR error behavior, but be relatively immune to changes in textual context. Our validation measures exhibit these properties.

5.1 Experimental Procedure

For our experiments, we used an on-line version of Herman Melville's novel, *Moby-Dick*. The ASCII text totals 1,179,194 characters. When formatted in 10-point Times at a spacing of 48 lines per page, the novel is 318 pages long. The OCR software we used was OCRServant, ver. 2.03, running on a NeXT workstation. For pages printed on a 400 dpi NeXT laserprinter and scanned at 300 dpi using a Ricoh IS410 flatbed scanner, the overall character recognition accuracy was about 99.8%.

We printed, scanned, and OCR'ed six complete copies of the novel: two each in Times, Courier, and Helvetica fonts. The error distribution was computed for each data set using the classification algorithm described earlier. We then analyzed all $\binom{6}{2} = 15$ possible pairings of the data sets using our four comparison measures.

We used the four similarity measures described previously: Bhattacharyya, Matusita, Vector Space, and Coin Bias. Table 2 compares all pairings of the six data sets (two copies of each of the three fonts), and also lists the minimum and maximum values attainable for each measure.

For the graphical presentation in Figure 2, each copy of *Moby-Dick* was divided into three equal-sized sections, resulting in eighteen data sets of slightly more than 100 printed pages each. We grouped the possible pairings ($\binom{18}{2} = 153$ in all) into four categories: same text-same font (ST-SF); different text-same font (DT-SF); same text-different font (ST-DF); and different text-different font (DT-DF). Since the "raw" distances are not directly comparable, we scaled and linearly translated them to fill the allotted space in the figure. For all the measures, it is possible to choose a threshold that differentiates between fonts when the same text is used. In all cases except for Matusita distance, the fonts can be differentiated reliably even when the text samples are different.

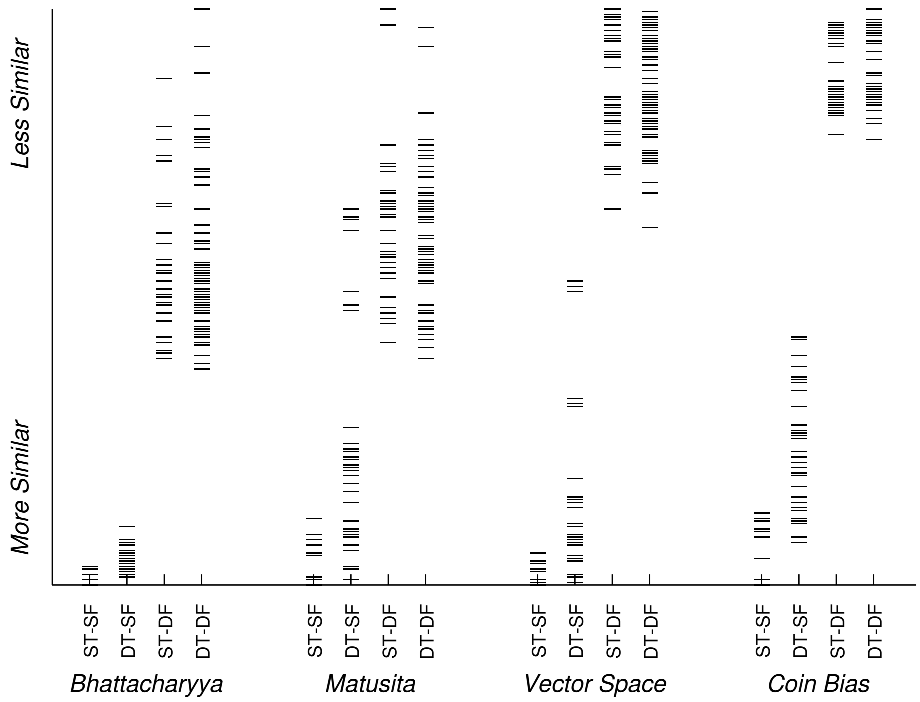

Figure 2: Font and text discrimination by validation measures.

How strongly correlated are the measures in terms of the rankings they induce? To examine this question, we computed the Spearman rank-order correlation coefficient, r_s, between each pair of measures:

$$r_s \equiv 1 - \frac{6\sum_{i=1}^{N} D_i^2}{N(N^2-1)} \tag{7}$$

where N is the size of the data set (153 in this case), and D_i is the difference in the ranking of the i^{th} element. As indicated in Table 3, the Bhattacharyya, Vector Space, and Coin Bias measures appear strongly correlated (but by no means equivalent), the Matusita measure significantly less so.

5.2 Issues in Sampling an Error Distribution

As Table 4 demonstrates, the number of different error patterns encountered is relatively small when compared to the total number of errors. For example, while there were 724 2:1 substitutions in the Times1 data set, only 33 of these were unique. (The most common 2:1 error, {`rn`→`m`}, occurred 331 times.) The errors are also sparse in their space; assuming an OCR alphabet size of 76 characters, there are roughly 1×10^{15} possible error types, only a small fraction of which will be seen in practice.

The question of whether to include non-error patterns (*i.e.*, $\{\alpha \rightarrow \alpha\}$) in the analysis is a natural one. When we augmented the data for Figure 2 in this way, all of the measures lost the ability to discriminate between fonts [25]. Since it seems reasonable to expect different fonts to exhibit different, characteristic OCR error patterns, this phenomenon requires some explanation.

The OCR software we used to generate the test data, in conjunction with the clean, first-generation copy, yielded very high accuracy rates, occasionally in excess of 99.8%. Thus, we can expect to encounter an error only every 500 characters or so. For different text samples, variations in the number of occurrences of each character are significant enough to overwhelm the effects of the OCR errors. Hence, different texts can be identified, but font-dependencies appear less significant.

Intuitively, we are attempting to approximate an error distribution using a sample. Even though 100 pages is large by the standard of some OCR experiments, our samples seem quite small when one considers that roughly half the non-zero entries occur fewer than ten times. By focusing specifically on error patterns, however, it becomes possible to induce a distribution that retains the relevant information for discriminating between two different fonts, or, hopefully, synthetic and real data.

As noted earlier, we only consider errors involving printing characters. Empirically, we found that including space errors yielded distributions with much higher variances, making it difficult to compare them reliably. This is undoubtedly a function of the fact that a single "mistake" made early-on during the top-down analysis of the page can result in a large number of otherwise independent-looking space errors (*e.g.*, a speck of dirt in the left margin can result in extra spaces being inserted at the beginning of every line). When space errors are included in the analysis, a slight bias can be noted towards same-font pairings for the Bhattacharyya and Coin Bias measures, but in no case are the classes linearly separable [25].

5.3 Comparison of the Measures

We now consider briefly how the measures relate to one another. Our first observation is that the Matusita measure is the only one that fails to distinguish between fonts (Figure 2). In a sense, this is not surprising. Matusita distance is simply the L_2 norm, and so has the property that $p_i = 0.01, q_i = 0.02$ contributes just as much to the final value as $p_i = 0.91, q_i = 0.92$, even though p_i in the former case occurs half as often as q_i, while in the latter case it occurs nearly as often. Thus, for comparing error distributions, a measure that depends solely on the difference between p_i and q_i may not be an appropriate choice.

The Vector Space measure exhibits a similar property. Consider the following three vectors:

$$v_1 = \begin{pmatrix} 4 \\ 1000 \\ 1000 \end{pmatrix} \quad v_2 = \begin{pmatrix} 5 \\ 1000 \\ 1000 \end{pmatrix} \quad v_3 = \begin{pmatrix} 4 \\ 1001 \\ 1000 \end{pmatrix}$$

From an information-theoretic standpoint, we would prefer that the added sample in the unlikely first row, as represented by v_2, would provide more information and yield a substantial difference from v_1. On the other hand, an extra sample in the much more likely second row, as represented by v_3, should give us less information, and result in a distribution that is more similar to v_1. However, the Vector Space method, which measures the angle between two vectors, does not have this property. As long as the element in the first row is small, the angle θ_{12} will be smaller than θ_{13} by a factor of $\sqrt{2}/2$.

The Coin Bias measure returns the probability that a knowledgeable observer will guess correctly based on a single sample. It is quite conceivable that two different pairs of distributions (p^1, q^1) and (p^2, q^2) could have the same coin bias. If the observer were allowed to see more than one sample from the secret distribution, the probability of guessing correctly might approach 1 very quickly for (p^1, q^1), but more slowly for (p^2, q^2). It is unclear whether a new measure could be devised that makes use of such a convergence property.

5.4 Evaluation of a Simple Defect Model

In addition to the previous experiments, we applied our validation measures to a simple document image defect model. In keeping with the theme of the paper, our focus is not on the model itself (*i.e.*, other more elaborate and accurate models have been described in the literature), but rather the process of validating it. For this test, we synthesized three types of distortion: smearing, smoothing, and thickening. Smearing was simulated by randomly repeating certain black pixels around character borders. Smoothing and thickening were implemented by averaging each pixel with its neighbors, using a threshold chosen to favor black pixels.

On visual inspection, the effects seemed quite plausible – we were unable to distinguish the real data from the synthetic. We then applied the methods of the previous sections for a more rigorous analysis. All of the measures displayed significant differences between the real and model-generated data. While both data sets exhibited self-similarity, none of real-synthetic pairings bore any resemblance ([25]). Thus, we were able to conclude that this particular noise model is not very accurate from an OCR standpoint. In practice, such a model might be abandoned at this point. Or, more likely, its parameters could be tuned to produce better results using continual feedback from the validation process.

6 Discussion

The most important property of the validation process is that it is grounded firmly in the error distributions induced by a particular classifier. Similar confusions on synthetic and real data is a *necessary* condition for the former to be substitutable for the latter. It is not, however, *sufficient*; a classifier may make identical mistakes on synthetic and real data for entirely different reasons.

Recall that the function $\mathcal{F}$ takes two error distributions and returns a single value quantifying their similarity. In practice, it will be necessary to alternate between attempts to improve the classifier and attempts to improve the defect model. A low value of $\mathcal{F}$ (*i.e.*, the distributions look similar) suggests improving the classifier. If the change in the classifier raises the value of $\mathcal{F}$, then the defect model needs to be tuned. This means that one can never entirely dispense with real data. Nonetheless, by validating a model based on its error distribution, we gain several advantages:

error distributions are universal Commercial classifiers are "black boxes": features, candidate rankings, and even confidence measures are not generally obtainable. However, every classifier returns the labels of the characters it claims to have recognized. These can be compared with the true labels to categorize the errors that have occurred.

what the classifier does not see, does not matter The approach we have proposed relieves the model from having to generate distortions that do not affect the classifier. For instance, if a classifier is inherently rotation invariant, then the error distributions, and hence $\mathcal{F}$, will not be affected by whether the model produces rotated samples or not.

known reliability Our confidence in $\mathcal{F}$ can be quantified by dividing the two data sets into subsets and computing the sample variance.

the data sets need not be identical If the character frequencies are different in the two data sets, the error distribution can be normalized.

rejects can be included If the classifier produces reject characters, they can be accommodated by incorporating additional entries in the error set.

we need not depend on a single classifier Classifier-dependence can be reduced by summing the error distributions from several classifiers, or by averaging the resulting values returned by $\mathcal{F}$.

To use error distributions, it is necessary to design and construct the error set carefully. Highly correlated errors introduce large variances that can invalidate the analysis (recall the example of the margin speck cited earlier).

OCR is most often a top-down process. While we have attacked the validation problem at the character level, much work remains at higher levels in the hierarchy. For example, a missed scan line that causes 20 errors on a line of text should be counted as a single, line-level event rather than as 20 strange-looking, high-variance-inducing character-level errors. If a baseline is computed incorrectly, the resulting error (*e.g.*, {`cow`→`COW`}) should probably not be counted as a multi-substitution (a 3:3, in this case). Likewise, the margin speck that results in extra spaces inserted at the beginning of every line should be counted as a single page-level event. Ideally, errors should be associated with the physical process that causes them, at the level where they occur.

7 Conclusions

Motivated by the appeal of synthetic data in OCR research, we have examined the notion of a defect model and its proposed applications. In doing so, we found that the link between synthetic data and the real world appears to be missing. The successful application of defect models depends critically on establishing such a link.

On the assumption that the training of classifiers is a key application of synthetic data models, we proposed a new necessary condition for defect-generator validation. This measure is based on the performance of the classifier under study on the types of data for which it was designed (*i.e.*, "real" data), and which the model should attempt to replicate. We believe that the proposed figure of merit, $\mathcal{F}$, measures exactly the right thing. Whatever the criterion (Bhattacharyya, Matusita, Vector Space, Coin Bias, etc.), $\mathcal{F}$ is easy to compute, and requires no assumptions other than the availability of samples from the two data sets to be compared and a reliable error classification procedure.

Such a simple check on the validity of synthetic data may go a long way towards alleviating skepticism about its potential usefulness in OCR research and development.

8 Acknowledgments

This paper is the synthesis of two independent works originally presented at the *Third Annual Symposium on Document Analysis and Information Retrieval* [26, 27].

George Nagy acknowledges the benefit of spirited discussions with Henry Baird, Tin Kam Ho, and David Ittner (AT&T Bell Laboratories), Junichi Kanai and Stephen Rice (UNLV), Michael Sabourin and Danny Thomas (BNR), Theo Pavlidis (SUNY-Stony Brook), Sharad Seth (UNL), and James Tien (RPI). He is also grateful for the financial support of the Northern-Telecom/BNR Educational and Research Networking Program.

Daniel Lopresti and Andrew Tomkins thank Jeffrey Esakov, Jonathan Sandberg, Jiangying Zhou, and the other members of the Carbon Group at MITL for many helpful discussions. Yanhong Li would like to thank Jonathan Hull (CEDAR/SUNY Buffalo) for his insightful comments and support.

All of the authors gratefully acknowledge the anonymous reviewers and their suggestions that helped make this a better paper.

The *Moby-Dick* text we used in our experiments was obtained from the Guttenberg Project at the University of Illinois, as prepared by E. F. Irey from the Hendricks House edition. The trademarks mentioned in this paper are the properties of their respective companies.

References

[1] S. V. Rice, J. Kanai, and T. Nartker. An evaluation of OCR accuracy. In *Annual Report of UNLV Information Science Research Institute*, pages 9–34, Las Vegas, NV, 1993.

[2] R. Bradford. Technical factors in the creation of large full-text databases. In *Proc. DOE Infotech Conf.*, Oak Ridge, TN, May 1991.

[3] H. S. Baird. Document image defect models and their uses. In *Proc. Int. Conf. on Doc. Anal. and Recog.*, pages 62–67, Tsukuba Science City, Japan, Oct. 1993.

[4] Q. R. Wang, Y. X. Gu, and C. Y. Suen. Applications of multi-layer decision tree in computer recognition of chinese characters. *IEEE Trans. Pattern Anal. Machine Intell.*, 5:83–89, 1983.

[5] T. Pavlidis. Effects of distortions on the recognition rate of a structural OCR system. In *Proc. Conf. on Comp. Vision and Pattern Recog.*, pages 303–309, Washington, DC, 1983.

[6] H. S. Baird and R. Fossey. A 100-font classifier. In *Proc. Int. Conf. on Doc. Anal. and Recog.*, pages 332–339, St. Malo, France, 1991.

[7] H. S. Baird. Document image defect models. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Analysis*, pages 546–556. Springer-Verlag, New York, NY, 1992.

[8] R. M. Haralick. English document database design and implementation methodology. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 65–104, Las Vegas, NV, Apr. 1993.

[9] F. Jenkins, J. Kanai, and T. Nartker. Using ideal images to establish a baseline of OCR performance. In *Annual Report of UNLV Information Science Research Institute*, Las Vegas, NV, 1993.

[10] S. Khoubyari and J. J. Hull. Keyword location in noisy document image. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 217–232, Las Vegas, NV, Apr. 1993.

[11] T. Kanungo, R. Haralick, and I. Phillips. Global and local document degradation models. In *Proc. Int. Conf. on Doc. Anal. and Recog.*, pages 730–738, Tsukuba Science City, Japan, Oct. 1993.

[12] M. Buchman. Distortion modeling for document images. In *Proc. DIMUND Work. on Page Decomp., Char. Recog., and Data Standards*, Harper's Ferry, WV, Aug. 1993.

[13] G. Nagy. On the auto-correlation function of noise in sampled typewritten characters. In *IEEE Region III Conv. Record*, New Orleans, LA, 1968.

[14] D. Lopresti, G. Nagy, P. Sarkar, and J. Zhou. Spatial sampling effects in optical character recognition. In *Proc. Int. Conf. on Doc. Anal. and Rec. (to appear)*, Montréal, Canada, Aug. 1995.

[15] K. Ishii. Design of a recognition dictionary using artificially distorted characters. *Systems and Computers in Japan*, 21(9):35–44, 1990.

[16] M. Parizeau and R. Plamondon. A handwriting model for syntactic recognition of cursive script. In *Proc. Int. Conf. on Pattern Recog.*, pages 308–312, The Hague, Netherlands, Sept. 1992.

[17] T. Kanungo et al. Document degradation models: Parameter estimation and model validation. In *Proc. IAPR Work. on Machine Vision Apps.*, pages 552–557, Kawasaki, Japan, Dec. 1994.

[18] R. A. Wilkinson et al. The first census optical character recognition systems conference. Technical Report NISTIR 1912, National Institute of Standards and Technology, Aug. 1992.

[19] H. S. Baird. Calibration of document image defect models. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 1–16, Las Vegas, NV, Apr. 1993.

[20] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *JACM*, 21(1):168–173, 1974.

[21] J. Esakov, D. P. Lopresti, J. S. Sandberg, and J. Zhou. Issues in automatic OCR error classification. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 401–412, Las Vegas, NV, Apr. 1994.

[22] P. A. Devijver and J. Kittler. *Pattern Recognition: a Statistical Approach.* Prentice Hall, Englewood Cliffs, NJ, 1982.

[23] D. Harman. Ranking algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 363–392. Prentice Hall, 1992.

[24] J. J. Hull and Y. Li. Word recognition result interpretation using the vector space model for information retrieval. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, Las Vegas, NV, Apr. 1993.

[25] Y. Li, D. Lopresti, G. Nagy, and A. Tomkins. Validation of image defect models for optical character recognition. Technical Report 69-93R, Matsushita Information Technology Laboratory, Sept. 1994.

[26] Y. Li, D. Lopresti, and A. Tomkins. Validation of document image defect models for optical character recognition. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 137–150, Las Vegas, NV, Apr. 1994.

[27] G. Nagy. Validation of simulated OCR data sets. In *Proc. Symp. on Doc. Anal. and Info. Ret.*, pages 127–135, Las Vegas, NV, Apr. 1994.

Bhattacharyya (range $[0,\infty]$)						
	Cour1	*Cour2*	*Times1*	*Times2*	*Helv1*	*Helv2*
Cour1	0.000	0.115	1.627	1.558	1.977	1.928
Cour2		0.000	1.602	1.535	1.898	1.848
Times1			0.000	0.093	1.374	1.312
Times2				0.000	1.335	1.314
Helv1					0.000	0.109
Helv2						0.000

Matusita (range $[0,\sqrt{n}]$)						
	Cour1	*Cour2*	*Times1*	*Times2*	*Helv1*	*Helv2*
Cour1	0.000	0.053	0.306	0.303	0.370	0.367
Cour2		0.000	0.273	0.270	0.341	0.339
Times1			0.000	0.030	0.294	0.292
Times2				0.000	0.309	0.305
Helv1					0.000	0.051
Helv2						0.000

Vector Space (range [1,0])						
	Cour1	*Cour2*	*Times1*	*Times2*	*Helv1*	*Helv2*
Cour1	1.000	0.990	0.228	0.225	0.034	0.037
Cour2		1.000	0.244	0.239	0.039	0.042
Times1			1.000	0.993	0.148	0.148
Times2				1.000	0.150	0.149
Helv1					1.000	0.979
Helv2						1.000

Coin Bias (range [0.5,1])						
	Cour1	*Cour2*	*Times1*	*Times2*	*Helv1*	*Helv2*
Cour1	0.500	0.596	0.930	0.927	0.978	0.977
Cour2		0.500	0.927	0.924	0.975	0.975
Times1			0.500	0.577	0.932	0.926
Times2				0.500	0.930	0.925
Helv1					0.500	0.598
Helv2						0.500

Table 2: Validation measures applied to real OCR test data.

	Bhattacharyya	*Matusita*	*Vector Space*	*Coin Bias*
Bhattacharyya	1.000	0.810	0.906	0.949
Matusita		1.000	0.774	0.778
Vector Space			1.000	0.954
Coin Bias				1.000

Table 3: Spearman rank-order correlation coefficients for all pairs of measures.

Total OCR Errors					
	$\rightarrow 0$	$\rightarrow 1$	$\rightarrow 2$	$\rightarrow 3$	$\rightarrow 4$
$0 \rightarrow$	n/a	619	1	1	8
$1 \rightarrow$	245	847	63	2	2
$2 \rightarrow$	0	724	270	25	7
$3 \rightarrow$	0	4	7	8	2
$4 \rightarrow$	1	0	0	3	2

Unique OCR Errors					
	$\rightarrow 0$	$\rightarrow 1$	$\rightarrow 2$	$\rightarrow 3$	$\rightarrow 4$
$0 \rightarrow$	n/a	6	1	1	3
$1 \rightarrow$	26	72	21	1	1
$2 \rightarrow$	0	33	44	15	4
$3 \rightarrow$	0	3	7	8	2
$4 \rightarrow$	1	0	0	2	1

Table 4: Error breakdown for the Times1 data set.