Validation of Document Image Defect Models for Optical Character Recognition*

Yanhong Li, Daniel Lopresti and Andrew Tomkins

Matsushita Information Technology Laboratory
Panasonic Technologies, Inc.
Two Research Way, Third Floor
Princeton, NJ 08540 USA
email: {yli,dpl,andrewt}@research.panasonic.com

Abstract

In this paper we consider the problem of evaluating models for physical defects affecting the optical character recognition (OCR) process. While a number of such 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 effectively indistinguishable from the errors encountered when using real scanned documents. We present two measures to quantify this similarity: the Vector Space method and the Coin Bias method. The former adapts an approach used in information retrieval, the latter simulates an observer attempting to do better than a "random" guesser. We compare and contrast the two techniques based on experimental data; both seem to work well, suggesting this is an appropriate formalism for the development and evaluation of document image defect models.

Topic areas: document image defect models, OCR error analysis, model validation

1 Introduction

Traditionally, researchers performing large-scale OCR experiments have employed one or more of the following approaches for generating test data:

1. **Real document images.** In this case, pages are printed and then scanned back into the computer. The data is undeniably authentic, but with an unfortunate limitation — the results are not precisely reproducible. Re-scanning a page multiple times yields a different document image, and hence different OCR errors each time. This natural variation is for the most part uncontrollable, complicating later analysis. Under this model, large-scale studies are an expensive, labor-intensive proposition.

2. **Public domain document image databases.** These make the same collection of canonical images available to a large number of researchers. This effectively addresses the problem of reproducibility, but by moving to the other extreme: the image for a particular page is fixed in advance and looks exactly the same every time it is processed.

3. **"Perfect" document images.** Here the pages are generated electronically, in memory, and are never rendered on a

[*] Presented at the *Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, April 1994.

physical medium (i.e., paper). These kinds of images are often referred to as *perfect* or *ideal*, meaning they exhibit none of the damage real-world documents experience. OCR software run on such images usually produces far better results than for scanned images. Because real documents are never perfect, this approach is somewhat artificial and can even result in anomalous behavior on the part of the recognition software (in one set of tests we performed, we encountered several inexplicable, unintuitive OCR errors that never seem to occur in practice).

4. **Document images generated through the use of defect models.** The limitations of the previous approach can be addressed by injecting simulated noise into perfect document images so that they more closely resemble real scanned data. Once a defect model has been determined, test images can be generated quickly, experiments become easy to control, and results gain reproducibility.

Several kinds of document defect models have been proposed recently [Bai90, Bai93a, KHP93]. Baird ([Bai90]) describes a parameterized model for local imaging defects; calibrations based on this model were investigated later ([Bai93a]). Kanungo, et al ([KHP93]) present a model for the perspective distortion that arises during the photocopying or scanning of thick, bound documents and for the degradation caused by perturbations in the optical scanning and digitization processes.

Nonetheless, informal experiments have shown us that although some models may generate images that look "real" to a human observer, they yield error patterns quite different from those seen when scanned pages are OCR'ed. Clearly, a measure for expressing how well a defect model simulates reality from the standpoint of OCR error behavior would have tremendous value. The development of a theoretical framework for validating models that provides a rigorous foundation for objective, empirical, and computable criteria for demonstrating their completeness was recently listed as an open problem by Baird ([Bai93b]).

In this paper we propose two such methods for validating document defect models. The first adapts a technique from information retrieval, the second takes a probabilistic approach. To examine the effectiveness of our ideas, we printed, scanned, and OCR'ed six different copies of Herman Melville's novel, *Moby-Dick*, yielding 1.2 Gigabytes of TIFF bitmaps (compressed), or 7 Megabytes of ASCII text. Our methodologies, and the results of these tests, are described in the sections that follow.

2 Defect Model Validation

Informally, we say that a document defect model is *validated* with respect to a given font and OCR package if it yields error patterns similar to those seen for real scanned images. To formalize this definition, we have developed two ways of quantifying the similarity between two sets of OCR error patterns:

Vector Space Method: For this measure we apply the *Vector Space model* from the field of information retrieval ([Har92]). We treat each OCR error pattern as a dimension in an error vector for the OCR output. The dot product of two such vectors gives the cosine of the angle formed by the vectors. The closer this value is to 1, the more similar are the two OCR outputs.

Coin Bias Method: In this procedure we implicitly model an observer watching the errors produced by an OCR process and "guessing" whether the original input was a real scanned image or a model-generated image. If the observer cannot distinguish between the two cases with sufficiently high probability, the defect model is deemed accurate.

These two approaches will be described in detail shortly, but first we point out that such similarity measures can be used not only to validate document defect models, but also to compare OCR results from different experiments involving scanned images. Since tests using real data are not precisely repeatable, similarity calculations between different sets of results allow us to quantify how much natural variation to expect.

Later we shall use this intuition to demonstrate the effectiveness of our validation procedures. In particular, we present OCR error characteristics for large amounts of data printed and scanned in three different fonts. One would expect that data drawn from related distributions, for instance identical pages of text printed in the same font and then scanned and OCR'ed, would exhibit similar error characteristics. We show that our validation techniques capture these similarities quite nicely. Likewise, our approaches detect that error characteristics drawn from unrelated distributions, for instance pages printed in two different fonts, should not appear to be similar.

The general framework we envision for document image defect model validation is presented in Figure 1.

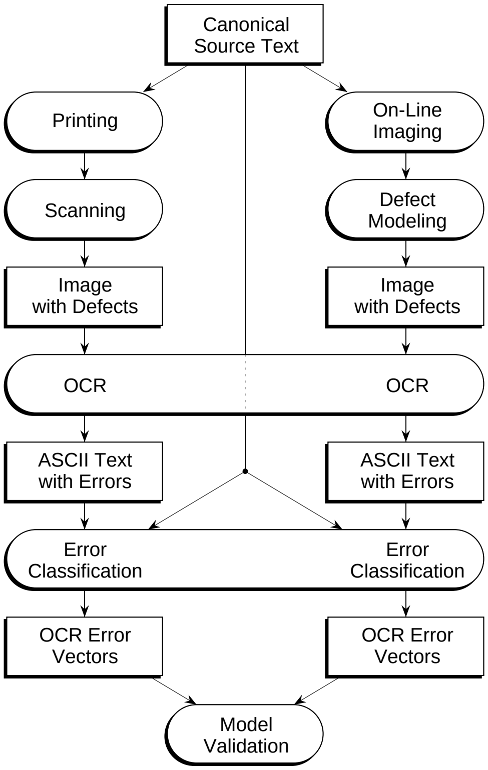


Figure 1: The document image defect model validation process.

As depicted in the figure, we start from an on-line reference text. In the case of the control, shown on the left side of the diagram, the text is printed and scanned, thus introducing noise. During printing, for instance, toner can "sputter" or the paper might jam briefly. Likewise, during scanning, uneven illumination, finite sampling, and binarization further distort the document image.

On the right side of the figure, 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 patterns is calculated by the validation procedure. If the value returned indicates a close enough match, the defect model is considered validated, otherwise it fails.

The next section presents our two approaches to quantifying OCR error-set similarity.

3 Validation Measures

The techniques described in this section each work independently based on OCR error characteristics. As a first step, the ASCII output is processed to identify any OCR errors that may have occurred. These are classified as belonging to one of six categories: deletions, insertions, 1:1 substitutions, 1:2 substitutions, 2:1 substitutions, and 2:2 substitutions. We use an approach based on the standard string edit distance model, modified to account for the special kinds of errors inherent to OCR, as described elsewhere [ELSZ94, Lop94]. Table 1 shows some examples from our experiments.

3.1 The Vector Space Method

The Vector Space Model is widely used in information retrieval to measure the similarity between two documents or between a query and a document [Har92]. In this approach, a document is represented by a vector of index terms or keywords. If an index term or keyword is present in a particular document, the corresponding element of the vector is set to 1; otherwise, it is set to 0. The similarity between a query and a document or between two documents is calculated by the inner product of the term vectors. Weighted vector inner products can be obtained through various term weighting methods. Such a vector matching operation, based on the cosine of the angle between vectors, is sometimes called cosine similarity.

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

Table 1: Example OCR error patterns.

$$\mathbf{sim}(\vec{V}^{(1)}, \vec{V}^{(2)}) = \frac{\vec{V}^{(1)} \cdot \vec{V}^{(2)}}{|\vec{V}^{(1)}||\vec{V}^{(2)}|}$$

The Vector Space Model has also been used recently in conjunction with OCR in a system that extracts keywords from an OCR'ed document and builds a vector whose elements represent the frequency of a particular keyword in the document. The document is then "classified" by searching through a database for similar keyword vectors, using weighted cosine similarity [HL93].

For our application, OCR error analysis and the validation of defect models, we assume that the total number of possible error patterns is n. An output set can then be represented by a vector $\vec{V} = \langle v_1^{(1)}, \ldots, v_n^{(1)} \rangle$, where the value of v_i is the number of errors of type i. That is, the weight of the i^{th} dimension is equal to the frequency of this error pattern in the test data. Then the cosine similarity between the two error vectors of interest $\vec{V}^{(1)}$ and $\vec{V}^{(2)}$ is calculated as:

$$\begin{aligned} \mathbf{sim}(\vec{V}^{(1)}, \vec{V}^{(2)}) &= \frac{\vec{V}^{(1)} \cdot \vec{V}^{(2)}}{|\vec{V}^{(1)}||\vec{V}^{(2)}|} \\ \text{where:} \quad \vec{V}^{(1)} \cdot \vec{V}^{(2)} &= \sum_{k=1}^{n} \left(v_k^{(1)} \cdot v_k^{(2)} \right) \\ |\vec{V}| &= \sqrt{\sum_{k=1}^{n} (v_k)^2} \end{aligned}$$

As an example, consider the error vectors shown in Table 2. Here $\mathbf{sim}(\vec{V}^{(1)}, \vec{V}^{(2)}) = 0.9749$. Note that characters recognized correctly are also considered an "error" in the case of 1:1 substitutions such as {a→a} or {b→b}. Doing so allows us to calculate the similarity between the original text, which contains no errors, and an erroneous OCR output.

3.2 The Coin Bias Method

As before, the goal is to quantify the similarity between two OCR error distributions. Again we shall assume we have two vectors representing the distributions over n possible error patterns:

$$\vec{V}^{(1)} = \langle v_1^{(1)}, \ldots, v_n^{(1)} \rangle$$

$$\vec{V}^{(2)} = \langle v_1^{(2)}, \ldots, v_n^{(2)} \rangle$$

The intuition for the Coin Bias method is as follows: suppose an observer has full information about the errors that result under each of the two distributions. Now say we secretly choose either distribution $\vec{V}^{(1)}$ or distribution $\vec{V}^{(2)}$ at random, generate OCR errors accordingly, and show them to the observer. The observer's "assignment" is to guess which distribution is being used to generate the data.

For instance, suppose $\vec{V}^{(1)}$ is highly likely to produce the error {a→o}, but $\vec{V}^{(2)}$ almost never does. An observer shown this error would probably be justified in guessing that $\vec{V}^{(1)}$ was being used.

For any particular error, the observer knows which distribution is more likely to make it and can guess accordingly. Therefore, we judge that the two distributions are similar if the observer can do little better than random at telling them apart. If the vectors are identical, for example, 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 now make this intuition more formal.

3.2.1 Formal Description of the Model

OCR error distributions are represented as n-dimensional vectors. Each element of the vector corresponds to a particular error pattern, and the value of the element is the probability of that error occurring. This differs somewhat

Pattern		a→a	a →o	b →b	b →B	m →m	m →rn	un →˜m	un →rm	
$\vec{V}^{(1)}$	⟨	8	2	8	5	5	5	2	1	⟩
$\vec{V}^{(2)}$	⟨	10	0	9	4	6	4	3	0	⟩

Table 2: Example of OCR error vector representation.

from our Vector Space method, where the value of the element is the frequency of the error. For example, a 2-dimensional vector $\langle 2, 6\rangle$ would be normalized to $\langle 0.25, 0.75\rangle$.

We make the simplifying assumption that the errors are independent. While this is not strictly true in real life, the correlations in the data are likely to be weak and so we should see a close approximation of the correct similarity value. At the expense of complicating the analysis, we could always make our error vectors more general to take local correlations into account.

Since we have assumed the errors are independent, we can summarize the probability of an observer guessing correctly by asking the following simplified question: if shown *one* error, chosen at random, what is the probability the observer will guess the distribution correctly?

An error of type i will be shown to the observer with probability $1/2(v_i^{(1)} + v_i^{(2)})$. The only reasonable strategy for the observer is to guess $\vec{V}^{(1)}$ if $v_i^{(1)} > v_i^{(2)}$ and $\vec{V}^{(2)}$ otherwise.

Next, note that the observer will be right when shown an error of type i with probability:

$$\frac{\max\{v_i^{(1)}, v_i^{(2)}\}}{v_i^{(1)} + v_i^{(2)}}$$

So the total probability p of the observer being right is:

$$\begin{aligned}
p &= \sum_{i=1}^{n} \Pr[(\text{correct guess on } i)] \\
&= \sum_{i=1}^{n} \left(\frac{1}{2}(v_i^{(1)} + v_i^{(2)}) \frac{\max\{v_i^{(1)}, v_i^{(2)}\}}{v_i^{(1)} + v_i^{(2)}} \right) \\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\max\{v_i^{(1)}, v_i^{(2)}\} \right)
\end{aligned}$$

3.2.2 Examples

Consider the two possible patterns $\{\mathtt{a} \rightarrow \mathtt{a}\}$ and $\{\mathtt{a} \rightarrow \mathtt{o}\}$. If the two vectors are:

$$\vec{V}^{(1)} = \langle 1, 0\rangle \qquad \vec{V}^{(2)} = \langle 1, 0\rangle$$

then we have $\mathbf{sim}(\vec{V}^{(1)}, \vec{V}^{(2)}) = 0.5$. Since the two distributions are identical, the observer has no advantage in guessing the answer and hence will be right half the time.

If, on the other hand, the vectors are:

$$\vec{V}^{(1)} = \langle 1, 0\rangle \qquad \vec{V}^{(2)} = \langle 0, 1\rangle$$

then the first distribution always maps `a` to `a` and the second distribution always maps `a` to `o`, so the observer knows with certainty which distribution produced a particular pattern. In this case, the probability returned is 1 and the defect model is deemed essentially useless.

As an intermediate example, consider the following:

$$\vec{V}^{(1)} = \langle 0.4, 0.6\rangle \qquad \vec{V}^{(2)} = \langle 0.35, 0.65\rangle$$

In this case, $\mathbf{sim}(\vec{V}^{(1)}, \vec{V}^{(2)}) = 0.525$, so the observer will guess correctly 52.5 times out of every 100, appreciably better than random.

We should point out one underlying assumption in the discussion so far. The Coin Bias method is Bayesian, so we have implicitly assumed that the data being used derives from real scanned images with probability 0.5, and from document defect models with probability 0.5. If these "prior probabilities" were different, the observer's strategy could be changed trivially to reflect this knowledge.

3.2.3 Multiple Trials

We have distilled the relationship between two sets of OCR error patterns to a single value p, the probability that an observer will guess correctly which process generated a randomly chosen error. But understanding the significance of the difference between two such similarity values, say 0.50005127 and 0.50089613, is difficult without some additional intuition. We propose the following technique to illustrate the bias in a more meaningful way.

The similarity p of two distributions gives the probability that the observer, on being presented with a single example, will guess correctly which distribution was used to generate it. So, in effect, the observer represents a coin with bias p and we can equate "flipping" the coin with seeing an OCR error pattern and guessing its source.

Suppose that instead of being shown a single example, the observer is shown a sequence of error patterns, and answers $\vec{V}^{(1)}$ or $\vec{V}^{(2)}$ after each one, according to the strategy just described. As we know, each guess is correct with probability p. Thus, after seeing 73 answers of $\vec{V}^{(1)}$and 27 answers of $\vec{V}^{(2)}$, we would logically assume that $\vec{V}^{(1)}$is the correct response. In this case, after seeing 100 error patterns, split 73-27, we can be fairly sure of our answer. We now ask the question: in general, how many patterns must the observer see to make a decision? For instance, if $p = 0.95$ only a few coin flips are required, while if $p = 0.50001$ we might very well see 100 coin flips and still have no idea which answer is correct. We now show how to determine the number of patterns the observer must see in order to be able to make a good guess.

Say a coin of bias p exists, but it is not known whether the bias is for heads or tails. The object is to determine on which side the bias lies. Clearly, if the coin is flipped 100 times and falls heads 51 times, the most reasonable guess is that the bias is towards heads. For a particular bias p, it is simple to calculate the probability that flipping the coin 100 times and voting with the majority will actually guess correctly.

The number of times the coin comes up heads will be distributed as a Gaussian centered, slightly to the right of center if the coin is biased towards heads, and slightly to the left of center if the coin is biased towards tails. We must assure that these Gaussians are disjoint for most of their area. Assume we flip the coin n times. If the bias is for heads, the number of heads is expected to be np, and we must show that if we travel k standard deviations from the expectation, we will not have crossed the "half-way point" to fewer than $n/2$ heads. So:

$$\begin{aligned} n(p - 1/2) &> k \cdot \sigma \\ n(p - 1/2) &> k\sqrt{np(1-p)} \\ n^2(p - 1/2)^2 &> k^2 np(1-p) \\ n &> \frac{k^2 p(1-p)}{(p-1/2)^2} \end{aligned}$$

We report the value of n to give a more concrete measure of the similarity between OCR error distributions.

In the previous section we gave an example of two distributions with $p = 0.525$. Using our visualization technique, this corresponds to 399 coin flips to be fairly certain we choose correctly. For a more accurate defect model, we might not be certain until we have seen 1000 coin flips; for a less accurate defect model, 100 coin flips might suffice.

We should point out that this value does *not* correspond to the amount of data needed to distinguish two distributions with high accuracy, as the result of each comparison is not a simple "yes" or "no", but a probability we are seeing a particular distribution. If this probability is 1.0, then clearly we are absolutely certain and need perform no further tests. This value corresponds to the situation in which the observer is allowed to guess $\vec{V}^{(1)}$ or $\vec{V}^{(2)}$, but is not allowed to include a confidence indicator along with the guess.

4 Experimental Results

We used an on-line version of the novel *Moby-Dick* by Herman Melville to generate page images which we then OCR'ed. The ASCII text occupies 1.2 Megabytes of storage, and when printed in 10-point Times fills 318 pages at 48 lines per page. The OCR software we employed was OCRServant running on a NeXT computer. Our basic character recognition accuracy was around 99.8% for "real" images produced by printing a page on a 400 dpi laserprinter and then scanning it back in using a Ricoh IS410 high-speed scanner at 300 dpi.

4.1 Error Analysis

We first investigated the similarity between error sets generated by OCR'ing different versions of *Moby-Dick*. Test cases were created by modifying several parameters of the printing/scanning process. Pages were printed in three different fonts: Times, Courier, and Helvetica. Each font was printed and scanned twice. Since a complete copy of the novel totals 318 pages, approximately 2,000 pages of 10-point characters were printed, scanned, and OCR'ed in all. Errors were classified into the previously discussed categories: deletions, insertions, and 1:1, 2:1, 1:2 and 2:2 substitutions. Our two similarity measures were calculated for each type of error. For the Coin Bias method, we computed both the probability p, and also the number of times a biased coin would have to be flipped in order to differentiate between the two sources with high probability.

Table 3 shows the similarity matrices for 1:1 substitutions for six versions of scanned *Moby-Dick* images in three different fonts. Times1 represents the first copy processed using Times, Times2 the second, etc. We shall employ this same naming convention throughout the remainder of the paper. The similarity between identical versions (e.g., Times1 and Times1) is always 1.0 for the Vector Space method and 0.5 for the Coin Bias method. Since the tables are symmetric, we need only show the upper half.

Because the OCR process is generally very accurate and we count correctly recognized characters along with the 1:1 substitutions, the similarity values are very close to 1.0 for the Vector Space method and 0.5 for the Coin Bias method. However, it is also quite evident that both schemes compute values several order of magnitude closer to these limits for test data based on the same font than for different fonts. This is precisely as expected.

We now consider the similarity matrices for 1:2, 2:1 and 2:2 substitutions for the six versions of *Moby Dick*. This data is presented in Tables 4, 5 and 6.

1:2 and 2:1 substitutions account for so few error patterns that different fonts may have nothing in common, thus we see numerous low similarity values in the tables. Different versions using the same font still have high similarity values, ranging in the Vector Space method from 0.55 to 0.99, except that 2:1 substitutions in Courier1 and Courier2 share no common error patterns. In fact, Courier1 has only two 2:1 substitutions: {`my`→`~`} and {`ux`→`w`}, while Courier2 has four patterns: {`Wh`→`~`}, {`ma`→⟨`sp`⟩}, {`nn`→`M`}, and {`o.`→`O`}.

2:2 substitutions are more informative; each font has an assortment of error patterns ranging from 24 (Helvetica1) to 68 (Times1). Times and Helvetica have a number of errors in common, but Courier errors are quite distinctive.

The overall similarity matrices (Table 7), combining the results for all types of substitutions, are dominated by the 1:1 case because of their much greater frequency.

Note that for the two most representative cases, 1:1 substitutions and overall substitutions, there are easily chosen thresholds that separate the results for a particular font from the results for all other fonts. For the Vector Space method, all similarity values greater than $1 - 10^{-6}$ are for the same font, and all other values are for different fonts. Likewise for the Coin Bias method, all distributions requiring over a million coin flips to differentiate are for the same font, and all smaller values are for different fonts. The fact that these classes are so well separated leads us to believe that both of our measures are appropriate for use in validation procedures, as outlined in Section 2.

4.2 Evaluation of a Defect Model

In this section we describe the results of applying our validation techniques to a simple document image defect model we have developed.

Skew, smear, blur, thickening, and thinning are some of the common defects observed during the printing and scanning process. Our approach is to develop a model for each kind of defect in isolation, then combine them together in a parameterized fashion. Individual defect models are often page based. For example, skew is a rotation of certain blocks in the document image; a block can be the whole page or part of the page (as might be caused by optical distortion or a misfeed during scanning). The same character at different locations on a page may suffer from varying amounts of distortion, of any of a number of forms.

Figure 2 shows an image generated by the smearing defect model.

We noticed that if we set the binarization threshold to the halfway point (128 in a [0,255]

Vector Space Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	1.0	0.99999994	0.999995	0.999995	0.999993	0.999993
Times2		1.0	0.999994	0.999994	0.999993	0.999993
Helvetica1			1.0	0.99999998	0.999997	0.999997
Helvetica2				1.0	0.999997	0.999997
Courier1					1.0	0.99999996
Courier2						1.0

Coin Bias Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	(∞) 0.5	(17m) 0.5001	(351k) 0.5008	(362k) 0.5008	(252k) 0.501	(248k) 0.501
Times2		(∞) 0.5	(338k) 0.5008	(347k) 0.5008	(249k) 0.501	(246k) 0.501
Helvetica1			(∞) 0.5	(58m) 0.5001	(673k) 0.5006	(611k) 0.5006
Helvetica2				(∞) 0.5	(676k) 0.5006	(603k) 0.5006
Courier1					(∞) 0.5	(27m) 0.5001
Courier2						(∞) 0.5

Table 3: Similarity matrices for 1:1 substitutions in three fonts.

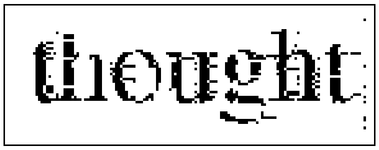


Figure 2: Smearing model effect.

range) during scanning, the bilevel image usually looked darker than the original, suggesting that more white pixels were changed to black than vice versa. To us, this effect seemed to be the only consistent, visible one when using small font sizes (e.g., 10-point). This observation motivated the development of our defect model, which adds three types of distortion: smoothing, thickening, and smearing. Smoothing and thickening were implemented by averaging each pixel by its neighbors with a threshold chosen to favor black pixels. Smearing was simulated by randomly repeating certain black pixels around the character border. Figure 3 shows a portion of a page generated by our model. Although this image may appear almost "perfect," it results in OCR accuracy rates comparable to those encountered using real data.

We tested our defect model using the validation techniques described earlier. For this experiment, we rendered *Moby-Dick* electronically in 10-point Times and passed the page images through our combined defect model. In the following we refer to this data set as *ModelT*. The base OCR accuracy was 99.7% — on visual inspection, the error patterns seemed to be reasonable. We now apply the methods of the previous sections for a more rigorous analysis.

From Table 8, we see that the defect model yields fairly reasonable results for 1:1 substitutions, but does less well for the other categories. For one obvious criterion, we would like the model to produce similarity values close to those calculated for two different printed versions of the document. A quick comparison of Tables 7 and 8 shows this is clearly not the case for the current implementation.

While this may sound discouraging from the standpoint of the simple defect model we described, the important point is that our validation procedures allowed us to quantify the performance of the model. The value of these methods extends beyond providing a simple "pass" or "fail" evaluation; they can be used to provide continual feedback as the model is tuned to produce more realistic results. We are now working on an improved version of our document image defect model.

Vector Space Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	1.0	0.866885	0.0	0.0	0.140160	0.163821
Times2		1.0	0.010784	0.0	0.0679067	0.043135
Helvetica1			1.0	0.551268	0.0	0.0
Helvetica2				1.0	0.0	0.0
Courier1					1.0	0.775826
Courier2						1.0

Coin Bias Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	(∞) 0.5	(11) 0.706	(1) 1.0	(1) 1.0	(4) 0.952	(4) 0.943
Times2		(∞) 0.5	(4) 0.994	(1) 1.0	(4) 0.982	(4) 0.969
Helvetica1			(∞) 0.5	(6) 0.836	(1) 1.0	(1) 1.0
Helvetica2				(∞) 0.5	(1) 0.999	(1) 0.999
Courier1					(∞) 0.5	(9) 0.733
Courier2						(∞) 0.5

Table 4: Similarity matrices for 1:2 substitutions in three fonts.

5 Conclusions

In this paper, we defined the defect model validation problem and proposed two different evaluation methods. We said that a document image defect model is validated if the OCR error patterns it induces are similar enough to those seen for real printed, scanned pages.

Both of our similarity measures are based on an error classification scheme that categorizes OCR errors as deletions, insertions, and 1:1, 2:1, 1:2 and 2:2 substitutions.

The Vector Space method considers each error pattern as a dimension of a vector that represents the recognition result. The inner product of two such vectors gives the similarity value between two sets of outputs. Normalized inner products generate similarity values in the range [0,1]: 0 for totally different results, and 1 for identical results.

The Coin Bias method, on the other hand, calculates the probability of successfully differentiating two recognition results, given a single error chosen at random from one of the two distributions. Such a "guess" is a probability based on the frequency of each error pattern in each error vector. If the overall probability is close to 0.5, it is difficult to differentiate the two recognition results; if the probability is close to 1, they are not similar. We also showed how to calculate the number of times the coin must be flipped in order to be certain which distribution is being presented — larger numbers of coin flips correspond to more similar distributions.

We tested our methods on a large corpus of data drawn from the novel *Moby-Dick*. The results of these experiments confirmed our intuition that the Vector Space and Coin Bias methods are reliable indicators of error pattern similarity, and may have significant value in a document image defect model validation procedure.

This research is continuing, both in the area of defect model development and validation, and in the in-depth analysis of OCR error patterns.

6 Acknowledgements

The authors would like to thank Jeffrey Esakov, Jonathan Sandberg, and the other members of the Carbon group at MITL for many helpful discussions. Thanks also to Jonathan Hull of CEDAR/SUNY Buffalo for his insightful comments and support. The *Moby-Dick* text we used in our experiments was obtained from the Guttenberg Project at the University of Illinois, as prepared by Professor E. F. Irey from the Hendricks House edition. The trademarks mentioned in this paper are the properties of their respective companies.

Vector Space Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	1.0	0.998564	0.003146	0.003919	0.0	0.0
Times2		1.0	0.000266	0.000729	0.0	0.0
Helvetica1			1.0	0.834058	0.0	0.0
Helvetica2				1.0	0.0	0.0
Courier1					1.0	0.0
Courier2						1.0

Coin Bias Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	(∞) 0.5	(242) 0.534	(4) 0.997	(4) 0.998	(1) 1.0	(1) 1.0
Times2		(∞) 0.5	(4) 0.999	(4) 0.999	(1) 1.0	(1) 1.0
Helvetica1			(∞) 0.5	(9) 0.741	(1) 1.0	(1) 1.0
Helvetica2				(∞) 0.5	(1) 1.0	(1) 1.0
Courier1					(∞) 0.5	(1) 1.0
Courier2						(∞) 0.5

Table 5: Similarity matrices for 2:1 substitutions in three fonts.

References

[Bai90] Henry S. Baird. Document image defect models. In *IAPR Workshop on Syntactic and Structual Pattern Recognition*, 1990.

[Bai93a] Henry S. Baird. Calibration of document image defect models. In *Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval*, 1993.

[Bai93b] Henry S. Baird. Document image defect models and their uses. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, 1993.

[ELSZ94] J. Esakov, D. Lopresti, J. Sandberg, and J. Zhou. Issues in automatic OCR error classification. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (to appear)*, 1994.

[Har92] Donna Harman. Ranking algorithms. In Ricardo Baeza-Yates William B. Frakes, editor, *Information Retrieval:Data Structures and Algorithms*, pages 363–392. Prentice Hall, 1992.

[HL93] Jonathan J. Hull and Yanhong Li. Word recognition result interpretation using the vector space model for information retrieval. In *Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval*, 1993.

[KHP93] Tapas Kanungo, Robert M. Haralick, and Ihsin Phillips. Global and local document degradation models. In *ICDAR 93*, 1993.

[Lop94] Daniel P. Lopresti. An algorithm for classifying optical character recognition errors. Technical Report MITL-TR-86-93, Matsushita Information Technology Laboratory, February 1994.

Vector Space Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	1.0	0.977272	0.720648	0.709578	0.001677	0.001484
Times2		1.0	0.633684	0.623868	0.001815	0.000674
Helvetica1			1.0	0.987835	0.005109	0.004112
Helvetica2				1.0	0.005662	0.004557
Courier1					1.0	0.972004
Courier2						1.0

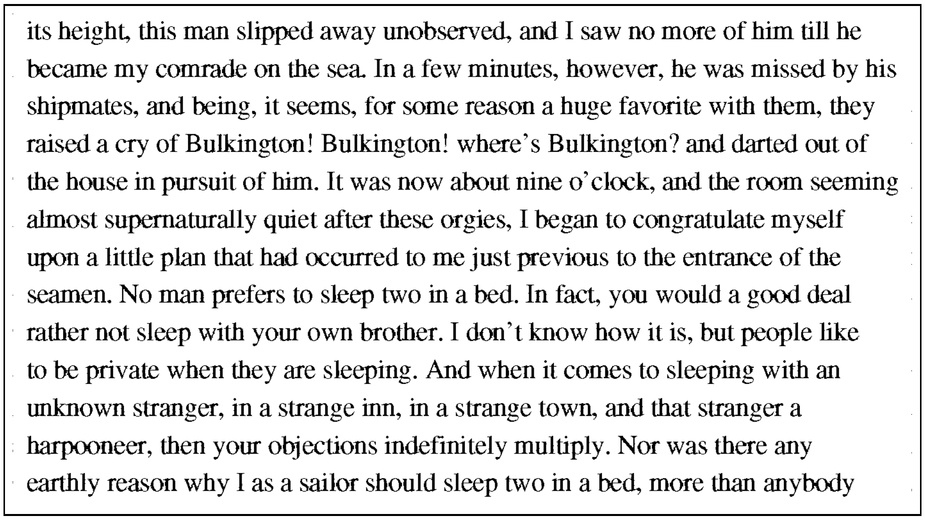
Coin Bias Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	(∞) 0.5	(24) 0.625	(6) 0.824	(6) 0.828	(4) 0.990	(4) 0.990
Times2		(∞) 0.5	(6) 0.831	(6) 0.835	(4) 0.993	(4) 0.994
Helvetica1			(∞) 0.5	(49) 0.583	(4) 0.986	(4) 0.987
Helvetica2				(∞) 0.5	(4) 0.989	(4) 0.990
Courier1					(∞) 0.5	(19) 0.644
Courier2						(∞) 0.5

Table 6: Similarity matrices for 2:2 substitutions in three fonts.

Vector Space Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	1.0	0.99999991	0.999994	0.999994	0.999992	0.999992
Times2		1.0	0.999993	0.999993	0.999992	0.999992
Helvetica1			1.0	0.99999997	0.999997	0.999997
Helvetica2				1.0	0.999997	0.999997
Courier1					1.0	0.99999997
Courier2						1.0

Coin Bias Method						
	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
Times1	(∞) 0.5	(6m) 0.5002	(215k) 0.5010	(220k) 0.5010	(163k) 0.5012	(158k) 0.5012
Times2		(∞) 0.5	(199k) 0.5011	(203k) 0.5011	(154k) 0.5012	(151k) 0.5013
Helvetica1			(∞) 0.5	(33m) 0.5001	(527k) 0.5007	(468k) 0.5007
Helvetica2				(∞) 0.5	(530k) 0.5007	(466k) 0.5007
Courier1					(∞) 0.5	(15m) 0.5001
Courier2						(∞) 0.5

Table 7: Similarity matrices for all substitutions in three fonts.



its height, this man slipped away unobserved, and I saw no more of him till he became my comrade on the sea. In a few minutes, however, he was missed by his shipmates, and being, it seems, for some reason a huge favorite with them, they raised a cry of Bulkington! Bulkington! where's Bulkington? and darted out of the house in pursuit of him. It was now about nine o'clock, and the room seeming almost supernaturally quiet after these orgies, I began to congratulate myself upon a little plan that had occurred to me just previous to the entrance of the seamen. No man prefers to sleep two in a bed. In fact, you would a good deal rather not sleep with your own brother. I don't know how it is, but people like to be private when they are sleeping. And when it comes to sleeping with an unknown stranger, in a strange inn, in a strange town, and that stranger a harpooneer, then your objections indefinitely multiply. Nor was there any earthly reason why I as a sailor should sleep two in a bed, more than anybody

Figure 3: Test document image with model-generated defects.

Vector Space Method						
ModelT	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
1:1	0.99999656	0.99999651	0.99999519	0.99999515	0.99999431	0.99999431
1:2	0.018203	0.004540	0.0	0.0	0.062432	0.107517
2:1	0.875737	0.878735	0.0	0.0	0.0	0.0
2:2	0.017742	0.002532	0.0	0.00539	0.002292	0.001640
all	0.99999567	0.99999550	0.99999461	0.99999456	0.9999937	0.9999937

Coin Bias Method						
ModelT	Times1	Times2	Helvetica1	Helvetica2	Courier1	Courier2
1:1 subs	(474k) 0.5007	(478k) 0.5007	(295k) 0.5009	(295k) 0.5009	(243k) 0.5010	(234k) 0.5010
1:2 subs	(4) 0.976	(4) 0.991	(1) 1.0	(1) 1.0	(4) 0.964	(4) 0.967
2:1 subs	(8) 0.757	(8) 0.753	(1) 1.0	(1) 1.0	(1) 1.0	(1) 1.0
2:2 subs	(4) 0.981	(4) 0.994	41) 0.999	(4) 0.999	(4) 0.991	(4) 0.995
all	(187k) 0.501	(178k) 0.501	(203k) 0.501	(204k) 0.501	(172k) 0.501	(163k) 0.501

Table 8: Similarity matrices for model results and scanned results.