Towards Whole-Book Recognition

Pingping Xiu & Henry S. Baird

Computer Science & Engineering Dept
Lehigh University
19 Memorial Drive West, Bethlehem, PA 18017 USA

E-mail: `pix206@lehigh.edu, baird@cse.lehigh.edu`
URL: `www.cse.lehigh.edu/~pix206`, `www.cse.lehigh.edu/~baird`

Abstract

We describe experimental results for unsupervised recognition of the textual contents of book-images using fully automatic mutual-entropy-based model adaptation. Each experiment starts with approximate iconic and linguistic models—derived from (generally errorful) OCR results and (generally incomplete) dictionaries—and then runs a fully automatic adaptation algorithm which, guided entirely by evidence internal to the test set, attempts to correct the models for improved accuracy. The iconic model describes image formation and determines the behavior of a character-image classifier. The linguistic model describes word-occurrence probabilities. Our adaptation algorithm detects disagreements between the models by analyzing mutual entropy between (1) the a posteriori *probability distribution of character classes (the recognition results from image classification alone), and (2) the* a posteriori *probability distribution of word classes (the recognition results from image classification combined with linguistic constraints). Disagreements identify candidates for automatic model corrections. We report experiments on 40 textlines in which word error rates fall monotonicaly with passage lengths. We also report experiments on an enhanced algorithm which can cope with character-segmentation errors (a single split, or a single merge, per word). In order to scale up experiments, soon, to whole book images, we have revised data structures and implemented speed enhancements. For this algorithm, we report results on three increasingly long passage lengths: (a) one full page, (b) five pages, and (b) ten pages. We observe that error rates on long words fall monotonically with passage lengths.*

Keywords: *document image recognition, book recognition, isogeny, adaptive classification, anytime algorithms, model adaptation, mutual entropy*

1. Introduction

We are investigating fully automatic methods for whole-book recognition. In [14] we introduced an information-theoretic framework for identifying significant disagreements between models—the *iconic* model and the *linguistic* model—and interpreting these as candidates for corrections of one or the other of the two models so that, when the updated models are reapplied to perform recognition, a lower overall error rate results.

In [14], a small-scale experiment, on a single textline, using the algorithm we now call ME1.0, showed that automatic corrections could be made to both models and that both character error-rates and word-error rates could fall. In this paper we report two experiments on longer passage lengths– in the first experiment, we do a series of scaling-ups to a full page based on ME1.0; in the second experiment, using a new algorithm ME2.0 introduced here, we've done experiments on one full page, five pages, and ten pages—in our drive towards whole-book recognition.

Our research builds on over a decades' work showing that adaptive classifiers can improve accuracy without human intervention[9]. Tao Hong[5] showed that within a book, strong "visual" (image-based, iconic) consistency-constraints support automatic post-processing that reduces error. These successes appear, to us, to be due largely to *isogeny*— the tendency of particular documents to contain only a small subset of all the typefaces, languages, image qualities, and other variabilities that occur in large collections[12]. It is well known that if models of the typefaces, languages, etc were known, even if only approximately, optimizing recognition jointly across all the models improves the accuracy[2, 7, 11].

In a long, highly isogenous book, identical (or similar) character images will occur multiple times, and the same word will also occur multiple times, independently. If the models are inaccurate, the errors caused by imperfect models will happen repetitively. The errors will also cause disagreements between the iconic model and the linguistic model; the overall model-disagreement on the whole passage is the accumulation of evidences for errors. A correct model adaptation (which leads to a better accuracy) will presumably lead to a lower overall model-disagreement, and an incorrect model adaptation will lead to a higher overall model-disagreement. By always checking it to validate the model adaptations, the recognition accuracy will quite likely to improve.

The principal goal of the work reported here is to understand the effects of scaling up the passage-length on which adaptation is attempted. To this end, we have redesigned the data structures, and rewritten inner loops, for higher speed. We have also been forced to face directly, for the first time, the implications of character segmentation errors (splitting and merging) since they are frequent in the errorful OCR results that our algorithm starts with.

In Section 2, we present the mathematical foundation of our framework, In Section 3, we analyze the previously described ME1.0 experiments, on larger passages than reported earlier, including an in-depth analysis of individual error cases and an analysis of performance curves . In Section 4, we introduce the design of ME2.0, including both the conceptual design and the implementation issues. We analyze the results of further scaled-up experiments using ME2.0 in Section 5, showing that under severe conditions (small training set, poor dictionary and many mis-segmentations), the word recognition rate for longer words increases significantly as passage length goes up.

2. Scaled-up Experiments Using ME1.0

2.1. Probabilistic Models

In our framework, two different kinds of models are required: an iconic model and a linguistic model. We impose four conditions on iconic models:

1. The iconic model, when applied to recognition, must compute *a posteriori* probabilities for all the character classes. (Of course, many such models are known [3]; we'll give details of our choice later.)
2. We expect that, in general, any given iconic model may be imperfect; however, we want it to be good enough to allow our mutual-entropy-based methodology to identify model contradictions and eliminate them. (We do not yet know exactly how accurate the model needs to be for this to happen reliably.)
3. An iconic model should be continuous in some image metric space: that is, it should give similar results on samples whose images are nearby one another under the metric. One example of such a metric is Hamming distance, but of course many others are known. Associated with this continuity assumption is the requirement that if one sample changes its *a posteriori* probability distribution among the classes, then image samples in its neighborhood should also be affected similarly. (We do not yet know how best to enforce these requirements.)

For a linguistic model, we expect to be given a lexicon (a dictionary containing valid words). The lexicon should cover most words appearing in the testing images, but may be incomplete. We also expect probabilities of occurrence to be assigned to each word in the lexicon: we can of course infer such statistics from a given corpus.

2.2. Independence Assumptions and Word Recognition

Now let X denote a sequence of T observations of character images (*i.e.* a word that is T characters long), and let S denote the true classes of these characters (in communication-theory terms, it is the inner state sequence that generates X):

$$X = (x_1, x_2, \cdots, x_T)\,, S = (s_1, s_2, \cdots, s_T) \tag{1}$$

where x_i are character images, and s_j are symbols of an alphabet. We adopt the following independence assumption, that each x_i is solely determined by its associated s_i:

$$P(x_i|s_i, \mathcal{F}) = P(x_i|s_i) \tag{2}$$

Where $\mathcal{F} = (Y, K)\,, Y \subseteq X - \{x_i\}$ and $K \subseteq S - \{s_i\}$. This assumption is similar to the one chosen by Kopec and Chou in their Document Image Decoding theory[6].

The *linguistic model* is $P(S)$, the prior probability of occurrence of any word S.

Our independence assumption implies that

$$\begin{aligned} P(X|S) =& P(x_1, x_2, \cdots, x_T|s_1, s_2, \cdots, s_T) \\ =& \prod_{i=1}^{T} P(x_i|x_{i-1}, \cdots, x_1, s_1, s_2, \cdots, s_T) \\ =& \prod_{i=1}^{T} P(x_i|s_i) \end{aligned} \tag{3}$$

By elementary definitions,

$$\begin{aligned} & P(x_1, x_2, \cdots, x_T) \\ =& \sum_{(s_1 s_2 \cdots s_T)} \left[P(x_1 x_2 \cdots x_T|s_1 s_2 \cdots s_T) \cdot P(s_1 s_2 \cdots s_T)\right] \\ =& \sum_{(s_1 s_2 \cdots s_T)} \left[\prod_{i=1}^{T} P(x_i|s_i) \cdot P(s_1 s_2 \cdots s_T)\right] \\ =& \alpha \cdot \prod_{i=1}^{T} P(x_i) \end{aligned} \tag{4}$$

where

$$\alpha = \sum_{(s_1 s_2 \cdots s_T)} \left[\prod_{i=1}^{T} P(s_i|x_i) \cdot \frac{P(s_1 s_2 \cdots s_T)}{\prod_{i=1}^{T} P(s_i)}\right] \tag{5}$$

Our *iconic model* (which provides posterior probabilities for all the classes) is denoted by the function $P(s|x)$ for all symbols s and all character images x. So we can derive $P(S|X)$, the result of word recognition informed by both the iconic and linguistic models:

$$P(S|X) = \frac{P(S,X)}{P(X)} = \frac{\left[\prod_{i=1}^{T} P(x_i|s_i)\right] \cdot P(S)}{\alpha \cdot \prod_{i=1}^{T} P(x_i)}$$
$$= \frac{1}{\alpha} \cdot \prod_{i=1}^{T} P(s_i|x_i) \cdot \frac{P(S)}{\prod_{i=1}^{T} P(s_i)} \qquad (6)$$

2.3. Mutual Entropy Measurements On Word Recognition

The *mutual entropy* $M(P, P')$ between the distributions $P(S|X)$ and $P'(S|X)$ is defined as:

$$\mathcal{M}(P, P') = -\sum_{S} P \cdot \log P' \qquad (7)$$

which measures the difference or "disagreement" between the two distributions $P(S|X)$ and $P'(S|X)$, where $P(S|X)$ is the *a posterior* probability distribution of the character string S given the image of the whole word X, and $P^{'}(S|X) = P(s_1|x_1) \cdot P(s_2|x_2) \cdot \dots \cdot P(s_T|x_T)$ is the distribution of the character string assuming that there is no linguistic constraints or the distributions of individual characters are independent with each other.

A property of mutual entropy which is critically important to us is that the more distributions P and P' differ from each other, the greater $\mathcal{M}(P, P')$ will be. Also, $\mathcal{M}$ can be further decomposed into the character-level disagreement measurements as follows:

$$\mathcal{M} = -\sum_{i=1}^{T} \sum_{s_i} P(s_i|X) \log P(s_i|x_i)$$
$$= \sum_{i=1}^{T} M\left(s_i|X, s_i|x_i\right) \qquad (8)$$

Where

$$M\left(s_i|X, s_i|x_i\right) = -\sum_{s_i} P(s_i|X) \log P(s_i|x_i) \qquad (9)$$

This measures the disagreement on individual character x_i. And $P(s_i|X)$ is the projection probability of $P(S|X)$ onto a particular element of a field.

$$P(s_i|X) = \sum_{s_j, j \neq i} P(S|X) \qquad (10)$$

If the iconic output "agrees" with the linguistic model, the two distributions should be close to each other, resulting in a smaller $\mathcal{M}$; otherwise, the linguistic information $P(S)$ will make $P(S|X)$ quite different from the iconic output $P(s_1|x_1) \cdot P(s_2|x_2) \cdot \dots \cdot P(s_T|x_T)$. As a result, *mutual entropy measures the disagreement between the iconic and linguistic models.* If the iconic models give out the correct answer but there is no corresponding entry in the dictionary, then the disagreement between the two model should be high, which results in a high value on $\mathcal{M}$ for that word.

$M\left(s_i|X, s_i|x_i\right)$ indicates disagreements between the *a posteriori* probability and the iconic probability for an individual character in the word. The disagreement for one character can be interpreted as a measure of the urgency of changing one model or the other. In order to change the iconic model, we can modify the $P(s_i|x_i)$ for that character's image. In order to change the linguistic model, we can modify the $P(S)$ for some word 'S'.

As a result, we have three different kinds of measurements:

1. The character-scale mutual entropy $M\left(s_i|X, s_i|x_i\right)$: this measures the model disagreements in regard to a specific character. It can indicate the urgency of changing the iconic model for that character.
2. The word-scale mutual entropy $\mathcal{M}$ measures the model disagreements in regard to a particular word. It can indicate the urgency of changing the linguistic model for that word.
3. The overall mutual entropy of the whole passage $\sum \mathcal{M}$: this measures the overall disagreements of the iconic model and linguistic model . We choose to use this as the objective function to drive improvements of both models.

So far, we've defined different measurements that operate at three different scales: character-scale, word-scale, and passage-scale. Do they have any relationship to the recognition rate? We argue that the overall mutual-entropy measurements (on the entire passage) are correlated with recognition rates.

1. If recognition performance is high, we expect small overall disagreement $\sum \mathcal{M}$. This is easy to understand: if character recognition is poor, either the iconic model has many errors, or the language model is incomplete: highly probably, they have a strong disagreement.
2. Within a word, if $\mathcal{M}$ is high, there are two possibilities: one, the word to be recognized may not in the dictionary; or, two, the word may contain incorrectly recognized characters due to an inaccurate iconic model.
3. For a single character, if $M\left(s_i|X, s_i|x_i\right)$ is high, there are two possibilities: one, the iconic model is wrong on this character; or, two, the language model may be incomplete.

Our strategy is to minimize these disagreements through a process of model adaptation: that is, applying a sequence of corrections to both models.

2.4. Algorithm Design

In this Section, we define both the iconic model and the linguistic model and describe the model-adaptation algorithm. The reader may wish to refer, to Appendix A for details (previously discussed in [14]) of the formal framework.

The criteria for designing the iconic model are: first, it should produce the probability $P(s_i|x_i)$; second, the character classifier for the iconic model should not be intrinsically complex, or from the perspective of statistical learning theory, the classifier's VC dimension should be low. This implies that, if a change to the iconic model affects one character image, it should impact all similar images; that is, changes to the iconic model should propagate to similar images.

For the iconic model, we have multiple choices: any character classifier with its complete configuration (training samples and parameters) can perform as an iconic model. In our approach, we use hamming-distance-based template matching as our classfication technique, for it is the simplest classifier we've ever thought of. For the classification process, the input character image is matched to the templates of each character code, getting the hamming distances. For hamming distances related to one character code, only the smallest distance are taken as the confidence value in regards to this character code. We compute a confidence value of the input image to each character code, and transform the confidence values into normalized values in probability sense. The normalized values should sum to one, and they must always non-negative. The formulae are as follows:

$$P(s|x) = \beta \cdot \exp(-\alpha \cdot \min_k d(x, T_s^k)) \tag{11}$$

$$\beta = \frac{1}{\sum_s \exp(-\alpha \cdot \min_k d(x, T_s^k))} \tag{12}$$

Where $P(s|x)$ denotes the normalized probability-like confidence value of a character code s given an input image; x denotes the input image to be recognized; T_c^k denotes the k-th template image of the character code s; and α is a manually-assigned parameter that determines the confidence of the classifier. α must be a positive real number, so that reducing the distance of the template matching increases the "probability" of that character code.

The α parameter of the classifier is important to the process of model adaptation: if α is assigned a greater value, the iconic model is more confident about its output because the contrast among the $P(s|x)$ increases; vise versa.

Now we say how we make changes to the iconic model. If the iconic model and linguistic model have high disagreement on one character, i.e. $M\ (s_i|X, s_i|x_i)$ is high, we can change the distribution $P(s_i|x_i)$ to more closely fit $P(s_i|X)$ and so to lower $M\ (s_i|X, s_i|x_i)$. We change a template of the top code c_{max} in $P(s_i|X)$ ($P(s_i = c_{max}|X) \geq P(s_i = c|X)$) to the image of x_i, in order to increase the ranking of the code c_{max} in the distribution of $P(s_i|x_i)$. For example, if we want a character's top candidate code to change from "b" to "h", we may change "h" 's template to this character's image. This increases the probability of the candidate "h" for this character, meanwhile lower down the ranking of "b" in the candidate list of the modified iconic model.

The linguistic model is a set of probability functions related to different lengths of words: $P^1(S^1), P^2(S^2), P^3(S^3), \cdots$, where $P^i(S^i)$ represents the language model with word length i. For each $S = \{s_1 s_2 \cdots s_T\}$ in the dictionary, $P(S)$ has a non-zero value. For each $S = \{s_1 s_2 \cdots s_T\}$ that is not in the dictionary, $P(S)$ equals zero. For example, the word "entry" should have a non-zero value in $P^5(S^5)$: $P^5(S^5 = entry) \neq 0$.

Thus model-adaptation algorithm is as follows: First, using the initial models, we recognize the entire test image and calculate $\sum \mathcal{M}$, $\mathcal{M}$ and $M\ (s_i|X, s_i|x_i)$. We randomly choose a character for model adaptation. We replace the template of the word-recognition-suggested code to this character image. If the overall ME increases, we revert our change; if the overall ME decreases, we adopt our change.

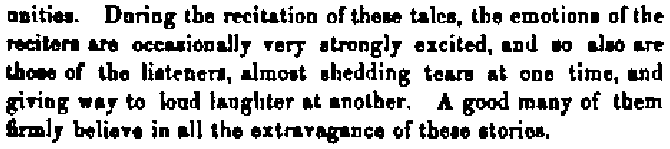
unities. During the recitation of these tales, the emotions of the reciters are occasionally very strongly excited, and so also are those of the listeners, almost shedding tears at one time, and giving way to loud laughter at another. A good many of them firmly believe in all the extravagance of these stories.

Figure 1. The top five lines of the testing page image

3. Scaled-up Experiments Using ME1.0

Our basic problem is, given an approximate *iconic* model (a model describing image formation and determining the behavior of a character-image classifier) and a *linguistic* model (a model describing word-occurrence probabilities), can one algorithm correct the deficiencies of both the iconic model and the linguistic model fully automatically, leading to a higher recognition accuracy?

In our last paper [14], we described our ME1.0 algorithm to deal with this problem. Briefly speaking, our model adaptation algorithm consists of many iterations, with each iteration consisting of two steps: modifying the models and validating the modifications. One potential way for modifying the models is to correct the models on characters or words that cause larger character- or word-level model disagreements. Validating the modifications is to check that the whole passage disagreements falls down by the model modifications. The Mutual Entropy, as a measurement of model disagreement, guides the automatic model adaptation operations to lower down the recognition error rate.

In [14], we tested our ME1.0 algorithm on a very small (a single-text-line) test case and showed that the model adaptations achieved a fully-correct result. Now two questions come: (1) could the algorithm apply to longer passages? (2) could the algorithm perform better on a longer and isogenous (in fonts, typefaces or image qualities) passage? So in this paper, we focused on discussions about the scaling issues of this technology. Our major goal is to find experimental proofs for our conjecture: the longer the passage length, the more accuracy we can achieve by our method.

We conduct a full page experiment using the roughly the same algorithm described in [14] (except forbidding the linguistic model adaptation) and demonstrate the monotonically improvements as the scale goes up.

At this stage, we do experiments on modifying the iconic model. Our policy is rather simple: pick up a character whose neighbourhood having a higher total ME, and try to make its image as a template of the code suggested by word recognition. (We only allow ONE template per character in this experiment.) If this action results in a lower overall ME on the whole passage, we adopt this change, otherwise we discard it.

This experiment consists of several scales of test with the same initial iconic model and the same initial linguistic model. Both models are not perfect at the very beginning. The iconic model is trained from a subset of that page image, with each character code attached with one template randomly chosen from multiple character image candidates.

The testing page is selected from page 28 of a book Volume_0000 in the Google Book Search Dataset [4]. Only five lines

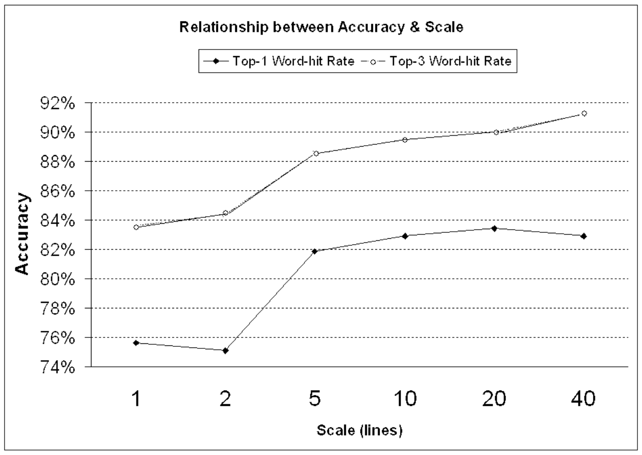

Figure 2. The relationship between the accuracy and the scale of the model adaptation range on a single page image.

are displayed in Figure 1.

The linguistic model is a word-list model, initialized from a dictionary that contains 45,000 words, the same kind of dictionary as used in Ispell program in Unix [8]. It is quite incomplete: the page image has many words that are infrequently used and therefore not appear in an ordinary dictionary. Also, in this experiment our algorithm takes the upper case and the lower case of a character as two distinct codes, so the words at the beginning of a sentence are usually the missing entries in the word-list linguistic model.

Now we show the performance curves in regarding to the passage-lengths of the model adaptation. Two curves are shown in Figure 2, one is for the top-one word-hit rate, the other for the top-three word-hit rate. We can see the roughly monotonically increasing tendency of the performances as the scale of the model adaptation range goes up. The top-one word-hit seems to saturate beyond the ten-line ranges, and we believe the reason is that the phenomenon is due to the limitation of single-template iconic model. A single template cannot match all variances of a character image on a larger scale so that the error rate of the iconic model tends to increase when the range goes up.

In spite of the limitation of the iconic model, we still see the monotonically improvements on the top-three word-hit rate. We interpret this phenomenon as the effect that even if the model adaptations cannot raise the true word codes to the top positions, they still try to improve the rankings of those true word codes.

In addition to the limitation of the iconic model, we also face the problem of the incomplete linguistic model. In this page image, all the words with the first letter being uppercase are missing in the language model because our linguistic model only have a lowercase representation for each word. The missing words in the language model counts for about ten percent of the total words in the testing image. It is quite promising that under such severe linguistic model deficiencies, our algorithm can still suggest corrections on the iconic model and shows the monotonic performance increasements as the passage-length goes up.

Table 1. The word "the" (word 3 of line 1)'s top-three rankings are changing as the model adaptation range goes up. The word matching scores are also listed.

Range	first	second	third
Line 1	(Ibn,1.86)	(Ike,-0.99)	(the,-3.50)
Line 1-2	(Ibn,0.07)	(Ike,-0.93)	(the,-3.25)
Line 1-5	(the,-0.27)	(rho,-0.97)	(Ike,-1.36)
Line 1-10	(the,0.64)	(rho,-0.66)	(she,-2.08)
Line 1-20	(the,0.61)	(rho,-0.97)	(she,-1.87)
Line 1-39	(the,0.14)	(rho,-1.07)	(she,-2.06)

In table 1, we demonstrate the increasing recognition performance on the word "the" in a series of incrementally enlarged tests. With the algorithm running on only the first line, the "the" word is misrecognized, being the third choice in the candidate list. When we extend the test to two lines (including line 1 and line 2) and run the algorithm again, the scores of the first and third candidate (the true candidate) are closer to each other. The big improvement comes after we extend the test to 5 lines (line 1 to line 5): the true candidate "the" was raised to the top candidate. We continue to extend the test, and the "the" remains at the first choice.

It is apparant that the isogeny is helpful on improving the iconic model, so that when more and more text samples of the same source are included in the test, the more evidences can be collected to guide the models to recognize "the" more accurately. In other words, if the model recognize wrongly on one "the", causing a larger ME measurement on the word, the increment of the whole passage ME (indicating the model imperfectness) is supposed to be amplified if there are more similar samples producing higher disagreements likewise.

Table 2 shows an example how errors on individual characters are corrected as the model adaptation range enlarges. The word "recitation" is seriously mis-recognized under the single-line model adaptation. As the model adaptation range increases, we see the gradual improvement of the iconic model output: the character recognition results have steady improvements from "rxci-InIloo", to "Γ33iIsIion", to "r33itation", to "re3itation", and correspondingly, the word ME monotonically decreases from 18.559 to 12.096.

In this test case, the increasing accuracy rate is correlated with the decreasing word-level ME. Also, in this test case, the correctness of an individual character classification is correlated with character MEs. Throughout this table, it is quite often that when an error on recognizing a character is corrected, the ME on that character reduces, and the ME on that word also reduces (see "t"'s and "i"'s ME changes). In the ten-line test, the third character "c" is misrecognized as "3", and the corresponding character ME is 1.942, which is the largest character ME in this test of this word.

Table 3 shows a special type of error that indicates potential model confusions, which is crucial to the function of the ME-based model adaptation. The groundtruth word "tale" is misrecognized as "isle", but the iconic model recognizes "t","a","l","e"

Table 2. The distribution of character-level MEs on the word "recitation" in experiments with increasing model adaptation ranges. As the range increases from line 1, to line 1-2, to 1-5, then to 1-10, the word recognition results for this word are "excitation", "recitation" , "recitation" and "recitation" respectively.

Range	1	ME	1-2	ME	1-5	ME	1-10	ME
r	r	4.0	Γ	2.4	r	1.7	r	1.3
e	x	1.4	3	2.8	3	1.9	e	1.6
c	c	0.9	3	2.7	3	2.7	3	1.9
i	i	1.1	i	1.2	i	1.0	i	0.8
t	I	2.0	I	1.6	t	1.1	t	1.0
a	n	2.6	s	2.4	a	1.1	a	1.1
t	I	1.8	I	1.4	t	0.8	t	0.9
i	l	1.3	i	1.2	i	1.0	i	0.7
o	o	1.1	o	1.3	o	0.9	o	0.9
n	o	2.3	n	1.5	n	1.7	n	1.8
Total		18.6		18.5		13.9		12.1

Table 3. A special type of error that indicates potential model confusions

	ME	code	prob.	code	prob.
t	3.032	t	0.402	i	0.030
a	2.426	a	0.333	s	0.096
l	1.754	l	0.357	.	.
e	2.228	e	0.216	.	.
tale	9.440	"isle"	0.165	"tale"	0.047

as their top-choice character recognition codes. Although "isle" is a wrongly recognized word, it has the potential to self-correcting because the large word-level ME explicitly indicates the code "i", "s" are causing confusions to the recognition.

Why the character recognition results are correct but the word recognition result is wrong, leading to a larger ME? When we compute the word posterior probabilities, we take into account the projection probabilities $P(s_i)$ of the character code at each character position (see equation 6). As "i","s" has much lower projection probabilities than that of "t", "a", the "isle" is more favored than the "tale" when calculating the postierior word probabilities.

The ME entropy will not necessarily reduce when the "t" and "a" dominates the character code posterior probabilites; it will only reduce when the scores of the confusion-causing code "i" and "s" are significantly lowered. It means that if "tale" is the correct word, the first character "t" should specifically NOT resemble the character "i", and the second character "a" should specifically NOT resemble the character "s". By studying this phenomenon, we can state that the ME measurements favor model adaptations that extend the differences between the words that tend to be confused with each other.

4. Description of Algorithm ME2.0

Motivated by the potential the experiments suggests, we conduct a larger scale experiment.

Many challenges exist for extending the scale of the experiment. The lacking of resources are the most serious problem we are facing. The resources include:

1. A large training set. The iconic model should be initialized by a passage image aligned with a transcript that is not necessarily perfect. But even we do not require the transcript perfect, we still have not managed to obtain it for various reasons.
2. A relatively complete dictionary. Though we believe that our algorithm has the potential to automatically fix the incomplete dictionary, we hope it is not too bad, at lease at this stage for startup. However, the dictionary we use is quite different from the book [4] so that the word recognition rate at the initial stage is quite low.
3. A reliable segmentation module. In the algorithm in Section 3, the algorithm cannot deal with segmentation errors, so we manually segmented the image. Of course we cannot depend on that because our goal is to design a practical system. Currently the only segmentation module available to us is the segmentation module in the platform of OCRopus [1] and we have to base our work on it.

We make some changes to ME1.0 to deal with these problems, and do a much larger scale experiment that shows the monotonic tendency of performance improvements on a ten-page level.

4.1. Conceptual Design

For the iconic model, we append some extra character images to one word image. The extra images are the combination of two adjacent individual character images. For example, the image of word “this”, should extends to a series of images “t”, “h”, “i”, “s”, “th”, “hi”, and “is”, if correctly segmented. Here, the images of “th”,”hi”,”is” are composed from the combination of the individual character images of “t”,”h”,”i”,”s”. Of course, the iconic model should at first have the codes “th”,”hi”,”is”, and be initialized with corresponding templates, so that ideally the model should output the combination codes “th”,”hi”,”is” for those additional images.

This technique helps to cope with the segmentation variations of a word image: if there is a undersegmentation, i.e. two adjacent characters touch each other and fail to be isolated, the individual character observations fails, but the combination image still gives a reliable iconic output; if there is an oversegmentation of a character, the individual observations fail because they cannot see the whole character, but their combination image forms an integral character image, which has the potential to be recognized correctly.

The algorithm should deal with the variations of the upper or lower cases of the first character in a word. This is a kind of redundancy that we would not favor: giving one word two instances in our data structure is too costly in regarding to either space or computation. We design a technique that transform this word-level redundancy to a character-level redundancy, therefore the algorithm saves time and space, and becomes much simpler.

We introduce a code into the code set of the iconic model for each upper-lower pair of characters. For example, we have a code “%a” representing both “A” and “a”, and have a code “%ab” for both “Ab” and “ab”. We call these codes *case-free* codes. We impose a rule about the matching score(in terms of a hamming distance) of the case-free codes, that a matching score is the minimal score of their original codes. For example, if “A”’s matching score is 50 and “a”’s is 100, then the “%a”’s score should be 50.

Except the matching score does not come from real matching, all other properties of the *case-free* codes are the same as normal codes, i.e. the probability of them together with all other codes should sum to one, and they can appear in the candidate list as other codes do. Now, if we write a word code as (“%w”,”i”,”t”,”h”,”%wi”,”it”,”th”), it represents for both “with” and “With”. Given the iconic output with probabilities of the case-free codes, the computation of the word posterior probability is the same algorithm as aforementioned.

4.2. Implementation Issues

As we all know, our methodology requires a book-scale processing during each iteration, so the speed of the algorithm is crucial to the feasibility of the algorithm. ME2.0 has implemented many techniques that make the scaling-up possible and feasible. Here are some techniques we implemented into ME2.0:

- A fast template matching technique.
  The image matching is always a time-consuming step, so we need to speed it up. We store the images in a compact way that one int type(4 bytes) represents 32 pixels’ values. When two images are compared, the corresponding image blocks are X-ORed to get a new differential block. We use the lookup table for counting the pixels of the differential blocks. Actually we do not need to “count” the pixels numbers: we pre-stored the number of pixels of each possible short int type number (16 bits) into a linear table, and just look it up when counting. Also, we only compair the intersection area of two images and get a matching score, and the pixels that are not in the intersection area all counts in the matching score. Again, the pixel numbers of each int-type image block are pre-stored and ready to look-it-up. According to some on-line notes, this technique can be at least ten times faster than just pixel-by-pixel counting.
- A hashing-table-based matching-score-lookup technique.
  When calculating the ME of a word, one of the most frequent operations is to lookup the matching score of a code. Since the codes of the iconic model are not well-ordered, we need to find a way to access them in a timely manner, otherwise it would take $log(n)$ time to locate a code if we use the c++ associate set utility in the Standard Template Library [13]. As a result, we use a hashing function to map a code to a memory address in a hashing table, and then we can access a code’s score directly without expensive operations to locate the code first.
- An efficient Mutual Entropy calculation algorithm.
  - The most expensive operation in calculating the ME is the expential function. When we calculate the exponential form of a matching score, we do not calculate the exponentials on-line: we pre-stored them in a lookup table because the number of possible matching scores are limited.
  - We do not consider any ranking when do ME calculating. This is because ranking the results needs to handle a large data structure for the sorting algorithm, which leads to an overhead. When calculating the whole book ME, this could significantly slow down the algorithm.

5. Further Scaled-up Experiments using ME2.0

The experiment results are reported as follows: We use only four groundtruthed text lines (150 characters in 45 words) from page 40 of the book [4] to initialize the iconic model. The linguistic model is initialized by the dictionary we used in ME1.0 (the previous algorithm). We use two templates for each character code (ME1.0 use only one for each). We do three tests to ME2.0: a test on one-page-level (page 40) model adaptation, a test on five-page level (page 40-44), and a test on ten-page level (page 40-49). We name them Test One, Test Two and Test Three separately.

On page 40, there are 1481 characters in 335 words. In this page, 269 segmentation errors happen, nearly one segmentation error per word. There are 212 words in this page that are longer than three characters, consisting of 1011 characters. And among these 212 words, 25 are missing in the linguistic model. In the following steps, these 212 words are our interested set on comparing the performance.

Our experiments show strong correlation between the increasing performance and the passage length: in Test One, only 38 words in the interested set are correctly recognized; in Test Two, 70 words are correctly recognized in this set; in Test Three, 82 words are correctly recognized in this set.

In Test Three, character accuracy rate is 58.7% on page 40. Of course this is not a high accuracy, but this is very promising considering our situation: small training set, poor dictionary, and poor segmentation.

The results of our algorithm suggest longer words are more easily to be recognized accurately if the word-segmentation errors are below two. However, we gave shorter words much higher weights than that of long words in calculating the total ME. As a result, the instability of the shorter words' ME were amplified and then caused the model adaptation on shorter words imperfect (fail to show the monotonic improvement).

6. Discussion and Future Work

This work has been motivated by the belief that isogeny will be more effective in driving recognition improvements as the passage length to be recognized increases[10]. The experiments reported here support that belief. We have seen that error rates on long words (greater than three characters) fall monotonically as passage length increases. We conjecture that the relatively small improvements on short words may result from our present policy of assigning short words higher weights.

Reponding to the challenges of mis-segmented OCR input, we have allowed redundant linguistic models (more than one per word), and our iconic models now apply to overlapping fields. These techniques are crucial for boosting the recognition performance and make the algorithm more powerful to survive in tough situations. What's more, in spite of this increased complexity, we have not seen a significant impact on runtimes.

Certainly our algorithm is sensitive to the accuracy of the initial (input) iconic and linguistic models, but we are heartened by having observed cases where even highly inaccurate initial models produce a rapid increase in accuracy.

We plan far larger tests, as soon as possible, scaling up rapidly to whole book-images. Currently our algorithm is in quadratic complexity of the testing word number: in each iteration, our algorithm tries iconic adaptation on each characters in the whole book, and then calculates the whole book's ME to verify the adaptation. However, we may reduce the algorithm complexity by limiting the model adaptations on hard-to-recognize characters and only updating the MEs of the words that are affected by model adaptations.

In the future, we hope to investigate these open questions:

1. Scaling up our experiments. In order to scale up from the current ten-page level to the whole-book level(typically 500 pages), some parallel computing technologies may be implemented.
2. Try and compare various policies for changing the models. The disagreement measurements only provides a framework, and there are various ways to implement a mutual-entropy-based auto adaptation system.

7. Acknowledgements

We are grateful for the crucial assistance of Luc Vincent of Google, Inc. in supplying the experimental data used in these experiments. Thanks particularly to Ahmad Abdulkader, Dar-Shyang Lee, Ray Smith and Suthaharan Sivasubramaniam for insightful comments on an earlier version. We have relied on the OCRpus software developed by Prof. Thomas Breuel. We wish also to acknowledge the continuing encouragement and advice of George Nagy, Prateek Sarkar, and Dan Lopresti.

References

[1] The OCRopus(tm) open source document analysis and ocr system, October 2007. The alpha release.

[2] T. Breuel and K. Popat. Recent work in the document image decoding group at xerox PARC. In *Proc., DOD-sponsored Symposium on Document Image Understanding Technology (SDIUT 2001)*, Columbia, Maryland, April 2001.

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification, 2nd Edition*. Wiley, New York, 2001.

[4] Google Inc. Volumn_0000, Book Search Dataset, August 2007. Version 1.0.

[5] Tao Hong. *Degraded Text Recognition Using Visual And Linguistic Context*. PhD thesis, 1995.

[6] G. Kopec and P. Chou. Document image decoding using Markov source models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–16:602–617, June 1994.

[7] G. Kopec, M. Said, and K. Popat. N-gram language models for document image decoding. In *IS&T/SPIE Electronic Imaging 2002 Proc. of Document Recognition and Retrieval IV*, San Jose, California, January 2002.

[8] Geoff Kuenning. Ispell(ver 3.1.20) word list, 1993.

[9] G. Nagy and H. S. Baird. A self-correcting 100-font classifier. In *Proc., IS&T/SPIE Symp. on Electronic Imaging: Science & Technology*, San Jose, CA, 1994.

[10] P Sarkar. *Style Consistency in Pattern Fields*. PhD thesis, Rensselaer Polytechnic Institute, May 2000.

[11] P. Sarkar, H. S. Baird, and X. Zhang. Training on severely degraded text–line images. [submitted to] IAPR Int'l Conf. on Document Analysis & Recognition, Edinburgh, August, 2003.

[12] P. Sarkar and G. Nagy. Style consistent classification of isogenous patterns. *IEEE Trans. on PAMI*, 27(1), January 2005.

[13] Bjarne Stroustrup, editor. *The C++ Programming Language*. Addison-Wesley, AT&T Labs, Murray Hill, New Jersey.

[14] Pingping XIU and Henry Baird. Whole book recognition using mutual-entropy-based model adaptation. In *Proc., IS&T/SPIE Document Recognition & Retrieval XII Conf.*, San Jose, CA, January 2008.