LANGUAGE IDENTIFICATION IN COMPLEX, UNORIENTED, AND DEGRADED DOCUMENT IMAGES

DAR-SHYANG LEE, CRAIG R. NOHL
Bell Laboratories, Lucent Technologies, Inc., 101 Crawfords Corner Rd. Holmdel, NJ 07733

HENRY S. BAIRD
Bell Laboratories, Lucent Technologies, Inc., 600 Mountain Ave. Murray Hill, NJ 07974

We describe algorithms for identifying the language of text in document images which are complex, unoriented, and degraded. We distinguish among seven languages: Chinese, English, French, German, Italian, Japanese, and Spanish. The page layouts may be *complex*, containing text blocks in unknown roughly Manhattan arrangements. The pages may be *unoriented*, that is, upright or rotated by 90, 180, or 270 degrees. The images may be *degraded* by digitization at coarse and unequal spatial sampling rates as in FAXes. We begin by segmenting the page into text lines in a manner oblivious to page skew and both page and text-line orientation. Then we distinguish between Asian and Latin scripts at any orientation. Chinese *versus* Japanese is decided at any orientation, and then their orientation is detected. On Latin scripts, we detect first orientation and then language. A variety of decision procedures are used, some hand-crafted (*e.g.* using spatial features and optical density distributions) and others trainable (*e.g.* using word unigram relative entropy models). Tests on 1088 standard (low) resolution FAX images show that our method accurately identifies scripts (98.16%), and language and page orientations (94.76%).

1 Introduction

The ability to identify automatically the language and/or script (writing system) of text in images of machine-printed documents has potential applications in OCR, information retrieval, routing, and indexing of the contents of digital libraries, document image archives, and FAX message traffic. In the last three years the first few attacks on this problem have been reported.

Spitz, Sibun, and Nakayama [1–3] have described several systems; in the most mature of these, Asian scripts are distinguished from Latin based on the distribution of upward concavities in text lines. Latin-script languages are distinguished by optical density distributions. Effects of layout complexity are not stressed in these papers, but variations in page skew and text-line orientation are tolerated. It appears crucial that the pages be correctly oriented; The report mentions only images scanned at normal to high resolution (300–400 pixels/inch (ppi)). Their methods are substantially hand-crafted, but some

decisions are automatically trainable.

Hochberg *et al.*[4] report a system for discriminating among 13 scripts (including Devanagari and Arabic). Their method is based on clustering size-normalized connected components and then matching them to language-specific templates. It handles complex layouts; in fact, no layout analysis other than connected-components extraction is needed. Skew up to ± 10 degrees is handled, and the method is insensitive to text-line orientation, but the report tacitly assumes that pages are correctly oriented (it may be possible to relax this successfully). Their method is substantially automatically trainable.

Wood *et al.*[5] briefly describe methods for script identification (among Latin, Cyrillic, Arabic, Han, and Hangul) using Hough transforms, morphological filtering, and analysis of density profiles resulting from projections along text-lines. How text-lines are isolated in complex layouts is not described. It is tacitly assumed that the pages are upright, and this assumption appears to be crucial. Facsimile images were used, but whether fine or standard resolution is not stated. No quantitative performance results are reported.

Sibun and Reynar[6] discuss language identification both in images of printed text and in passages of encoded text. They show relative entropy can effectively distinguish twenty seven Roman-alphabet languages on short passages and requires very little training data. The issues of complex layout and image degradation are not discussed in the paper. The pages are assumed to be correctly oriented. A more detailed discussion of their methods can be found in Section 8.

We have attempted to extend language and script identification to cases significantly more challenging than previously reported, combining complex layouts, unoriented pages, and degraded images. We report initial progress towards this goal, achieved on a small number of commercially important scripts and languages. We plan to extend our system soon to many more scripts and languages.

2 Overview of the System

One intuitively appealing strategy, which other researchers have followed with success, is to decide first the script and then the language. However, faced with complex layouts, unknown orientations, and coarse and unequal digitizing resolutions, we were unsure that this was the correct course. We asked ourselves, is it easier to detect script when ignorant of orientation, or orientation when ignorant of script? For Asian-script languages, is it easier to detect language when ignorant of orientation, or the reverse? Does the same strategy hold for Asian-script and Latin-script languages? Questions of this sort, for which the

literature provided little guidance, dominated the early stages of our research and prompted a long series of experiments which, for lack of space, we cannot recount in detail here.

The best results we have achieved to date requires the following strategy. First, the page image is skew- and shear-corrected and segmented into blocks and lines of text, using algorithms [7] that do not require prior knowledge of page skew or page and text-line orientation. After this, all decisions are based on statistics of features extracted from isolated text lines, considered separately (not on characters, words, or text blocks). It is thus possible, in principle, for us to identify language within a single text line. However, for greater robustness, we will, in this report, combine decisions (or statistics) from all the text lines on the page to support a final decision at the page level.

The order of execution is as follows. Within a page, we visit each text line (in arbitrary order), extracting features and computing statistics for each (details of these will be given below). Each text line is immediately classified as Asian or Latin (or rejected), in a way that requires no prior knowledge of the orientation. Feature statistics which support other decisions are accumulated and, at the end of the page, the remaining decisions are made in the following order. First, we distinguish between Asian and Latin scripts, determined by a majority vote of the text-line classifications already decided. If the page's script is Asian, we then decide the page's language using feature statistics accumulated from all text lines (again, at any orientation); then, using knowledge of the Asian-script page's language, we detect the page's orientation. If the page's script is Latin, we detect the page's orientation first and then decide the page's language. An overview of this process is shown in Figure 1.

Details of the methods used for each of these decisions, along with test results, are discussed in the sections that follow. But first, we briefly summarize the training and test data used.

3 Training and Testing Data

No satisfactorily large and diverse data base of systematically degraded and rotated images of multilingual documents complex layouts were publicly available when we started this project, so we collected our own. Table 1 indicates the number of original pages we collected.

For each original page, we acquired (using FAX equipment), two images: one at 200x100 ppi (FAX "standard" resolution), and one at 200x200 ppi (FAX "fine" resolution). We then rotated each fine-resolution image by 90, 180, and 270 degrees (perfectly in software, not by rescanning). These were subsampled in the vertical direction to derive a set of rotated standard-resolution images.

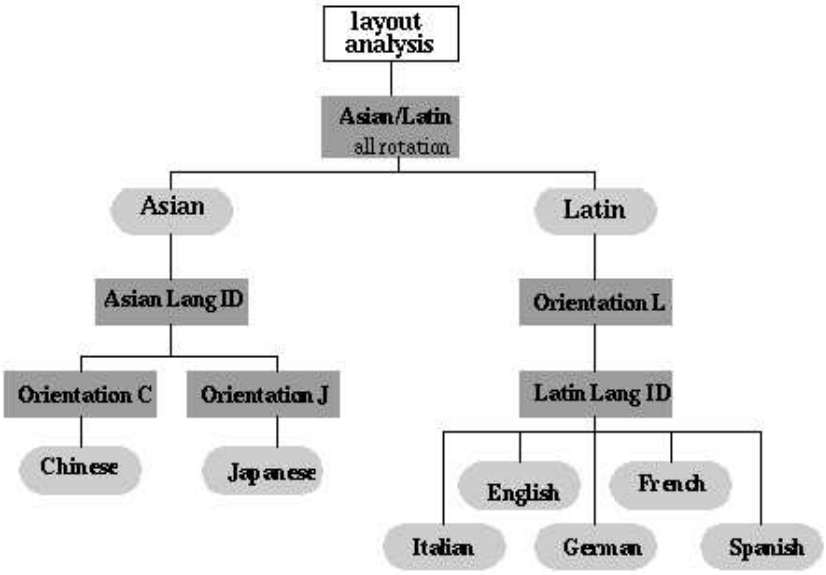

Figure 1: System overview.

In this way, each original page contributed eight images representing both FAX resolutions at each of the four orientations.

4 Distinguishing between Asian and Latin Scripts

Our first decision is to discriminate between languages in Asian script (Chinese and Japanese) and languages in Latin (European) script (English, French, German, Italian and Spanish). We assume no prior knowledge of the page orientation, but we do assume that the bulk of the text is in a "Manhattan" layout, that is, with text lines running either horizontally or vertically (after skew and shear correction). We possess reliable methods [7] for detecting text-line orientation without prior knowledge of page orientation, and without recognizing individual symbols by shape; therefore we compute text-line orientation before identifying the script.

Spitz[1] has shown that Asian and Latin scripts can be discriminated using a set of spatial features based on optical density and distribution of upward concavities. We generalized his features to work effectively at any of the four

Table 1: Number of original pages collected, by language and by training and test sets.

Language	*Train*	*Test*
Chinese	22	23
English	43	42
French	50	50
German	47	47
Italian	21	22
Japanese	61	61
Spanish	27	27

orientations. In the following discussion, for brevity, we will speak as if the text line were always horizontal, but it should always be clear in context how the feature may be analogously computed for vertical text lines. The contents of a text line at this stage is a set of black 8–connected components, grouped roughly into "characters" by a rule sensitive to overlaps perpendicular to the text line direction (see [7]). These "characters" do not always correspond 1–1 to correctly isolated symbols. The following features, also shown in Figure 2, are computed:

- position of concavities: horizontal positions of upward- and downward-facing concavities (as defined by Spitz);
- distribution of character height, relative to text-line height[a];
- character bounding-box top and bottom profile; and
- optical density (as defined by Spitz).

In practice we observe that these features do not vary greatly when the page rotates to the four orientations. This holds for reasons that are different for Latin and Asian scripts. For the Latin script, the features are always computed from the top and bottom of characters, whatever the page orientation may be. This is not true for the two Asian scripts: for example, text lines in a Japanese newspaper page may run both horizontally and vertically even though all individual characters are upright. Thus, for the two Asian scripts, the features may be computed from the top/bottom or left/right of the characters, depending on the orientation of the characters with respect to

[a] Text-line height is the height of the text-line bounding box (or, of course, width for vertical text lines).

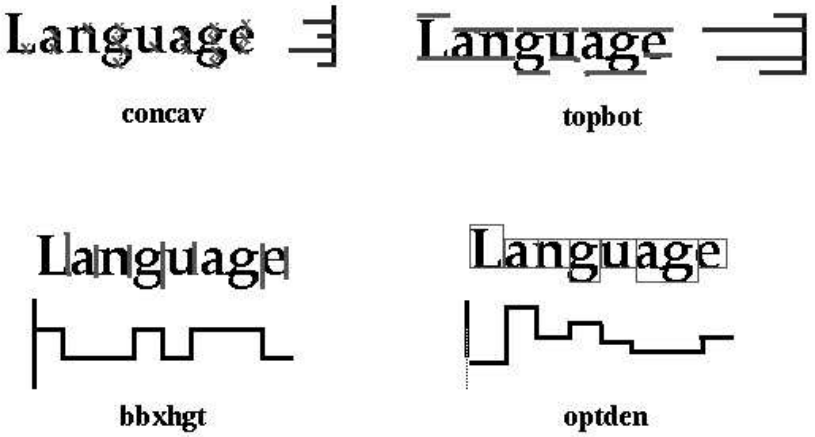

Figure 2: Four types features extracted for script identification: upward and downward concavities (concav), character top and bottom profile (topbot), distribution of character height (bbxhgt) and optical density (optden).

the text-line direction; fortunately, both Chinese and Japanese characters are "symmetric" in the sense that these features remain fairly constant whether computed from the top/bottom or from the left/right. These features, when accumulated over a page, show great contrasts across Asian and Latin scripts. See Figure 3.

The decision is a voting procedure with inputs from fixed thresholds on the mean and variance of feature distributions, carried out as follows. Each of the four feature distributions is normalized by scaling by the text-line height. Four independent decisions are made based on each feature, as follows.

1. For the concavity feature, if the total black area contained in the two ranges 0.2 to 0.5 and 0.6 and 0.8 is greater than 0.8, we decide Latin. If the area falls between 0.6 and 0.8, we decide Asian. No decision is made if the area is less than 0.6.

2. The mean μ and variance σ^2 of the character-height distribution is computed. If the discriminant $(\mu * 0.2) + \sigma^2 - 0.16$ is positive, we decide Asian, otherwise Latin.

3. For the character bounding box alignment, the variance alone is used. If it is greater than 0.12, we decide Asian, otherwise Latin. Since Asian characters generally have equal size, character bounding boxes align at

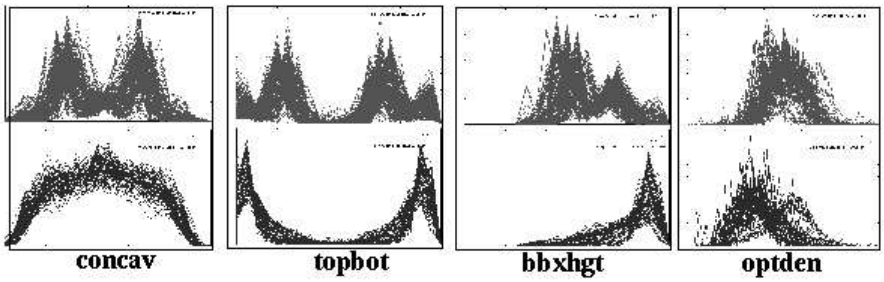

Figure 3: Comparison of features, normalized to textline height, for Asian (bottom) and Latin (top) script. Each line represents features accumulated over a page, plotted across pages of all orientations.

the top and bottom of the text line, resulting in a larger variance than Latin characters, whose bounding boxes tend to align near the baseline and x-height line.

4. The optical density distribution contributes another discriminant $(0.05 * \mu) - \sigma^2 - 0.015$: if it is positive, we decide Latin, else Asian.

Majority vote among these four decisions gives the decision for the text line. While a few text lines — especially short lines — may be classified incorrectly, usually a large majority on the page are correct, so voting among the text lines gives a reliable decision for the page: 98.1% correct on standard resolution images and 99.6% correct on fine resolution.

5 Identifying Language within Asian Scripts

In our initial experiments, we could find no single reliable method for detecting the orientation of both Asian languages. We speculated that language-specific characteristics of Chinese and Japanese might allow us to determine orientation once the language is known. Thus we developed a set of rotationally invariant features to be used for Asian language determination, namely the density of ink and number of ink runs in rows or columns perpendicular to the text-line orientation, with special normalization to reduce their sensitivity to typeface style and imperfections in character segmentation.

Spitz reports that the distribution of "cell densities" supports good discrimination among Chinese, Japanese, and Korean. We adopted his method with a slight modification. We first compute the vertical projection profile of text lines (as usual, we describe only the horizontal text-line case), with

all zero entries (white gaps) eliminated. The resulting compressed profile is divided into "cells," square regions with sides equal to the text-line height.

We define the "density" of a cell to be the fraction of its area that is black, multiplied by the average number of runs per cell computed over the page. Thus, for example, a cell containing a single stroke 10 pixels thick will be assigned a lower density than a cell containing 5 strokes each 2 pixels thick. This normalization policy is intended to improve invariance with respect to character complexity, and was motivated by the following observations. Chinese characters usually have a higher density than Japanese characters, due to the complexity of the characters. However, some Japanese typefaces use thickened strokes for Kana characters, raising their raw cell densities. (As an illustration, consider that a thin 'E' may not have more black pixels than a thick, bold-faced 'I'.) To compensate for this, we multiply the density by the average number of runs/cell, assigning thin but complex character a higher cell density.

The mean μ and variance σ^2 of the set of densities (for the text line) are computed. The 2-D distribution of (μ,σ^2) pairs for all text lines in the training set was analyzed and a linear discriminant inferred to separate Chinese from Japanese. Majority vote among the text-line decisions gives the page decision.

When tested on fine-resolution FAX images, this method correctly identified 100% of the Japanese pages and 98.9% of the Chinese pages. On standard-resolution images, it correctly identified 99.6% of the Japanese pages and 93.5% of the Chinese pages.

6 Detecting Orientation of Chinese/Japanese

We devised a set of features for language-specific orientation detection. Compared to Latin scripts, orientation detection in Asian scripts is somewhat more difficult since, as we mentioned earlier, character orientation is not determined by line orientation.

Our approach is based on the observation that "/\" is a commonly occurring stroke pattern in both languages. However, automatically locating all such patterns would, we fear, be time consuming and error-prone due to the degraded image quality. Thus we explored an alternative set of features based on character chain codes, which proved to be efficient and effective. All pairs of adjacent edge pixels in the image are encoded as representing one of four local edge directions: horizontal, vertical, left diagonal ("/"), and right diagonal ("\"). Each left diagonal chaincode is encoded by +1 and right diagonal by -1. Horizontal and vertical edges are ignored. Two features are introduced. The *topness* feature flips the signs of the diagonal edges on the right side of

the image along its bisector, and sums the edge values over the entire image. If the image is facing up, we expect the *topness* feature to be strongly positive. This can be conceptualized as folding the right half of the character image to the left half, changing all right diagonals to left diagonals. Consequently, this yields a high left diagonal count if "/\" is present, indicating the image is facing up. This is illustrated in Figure 4. On the contrary, if the character image faces down, a high right diagonal count is expected after folding due to the presence of "\/". A *leftness* feature is similarly defined: folded along the horizontal bisector to determine if the image faces left or right. Notice that when the image faces up or down, the left and right diagonals are roughly equal after folding, resulting in a small value for the *leftness* feature.

Character orientation is determined in two steps. First, the percentages of edges in the horizontal and left diagonal directions are used to determine if characters are facing either "up/down" or "left/right." If a character faces "up/down," the *topness* feature is used to determine if it faces up or down; otherwise, the *leftness* feature is used to determine if it faces left or right. For both Chinese and Japanese, in upright characters, horizontal strokes are more common than vertical strokes. This is used to confirm the correctness of the first step, as follows. If a page has more horizontal than vertical runs, indicating "up/down" orientation, and this contradicts the decision made by the above rule, the page is rejected. On fine-resolution test images, this multistage decision process detected the correct orientation in 98.9% of the Chinese pages and 96.7% of the Japanese pages.

On standard-resolution test images, we experimented with a single linear discriminant function which is a function of nine features: the ratio of horizontal to vertical runs, the left diagonal using vertical folding, diagonal using horizontal folding, percentage of edges in each of four direction, and mean and variance of the cell density (used for script discrimination). This method achieved, surprisingly, better results on standard resolution than we were able to achieve on fine resolution: 100% correct on both Japanese and Chinese.

7 Detecting Orientation of Latin Script

Latin script orientation detection is constrained, helpfully, by knowledge, determined earlier, of the text-line orientation. So, for example, if the text line is horizontal, we need to distinguish between only upright and upside-down character orientations.

To determine the orientation of a line of Latin text, we estimate the loca-

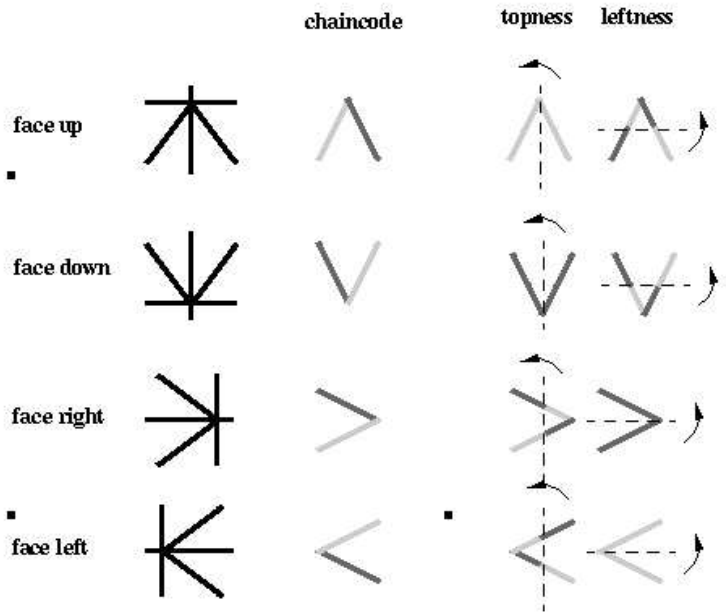

Figure 4: *Topness* and *leftness* features for Asian language orientation detection. Left diagonal edges, in lighter gray, are encoded as +1, and right diagonal edges, in darker gray, are encoded as -1.

tion of two reference lines: the baseline [b] and the x-height [c] line.

We have found there are usually more "marks" above the x-height line (in the "ascender region") than below the baseline (in the "descender region") due to the fact that ascenders are more common than descenders as well as the occurrence of capital letters, diacritical marks and accents in the ascender region. By extracting the x-height line, the average number of marks in these two regions can be used to determine the correct orientation of a Latin page, as follows. For each text line, the difference between the average number of runs in the rows above the x-height line and the average number of runs in the rows below the baseline line is computed. If this value, accumulated over the entire page, is positive, indicating more runs above the x-height line, then we decide that the page's orientation is upright, otherwise upside down. As usual, majority vote among the text-line decisions give the page decision.

This method proved to be 98.1% correct on the fine-resolution test set, and 96.0% correct on the standard-resolution test set (each test set contained 744 page images: 186 pages at 4 orientations each).

8 Identifying Language within Latin Script

Once the page is determined to be printed predominantly in the Latin script, we proceed to identify the language — among English, French, German, Italian, and Spanish — as follows. A polyfont OCR subsystem [8], trained on a superset of all five languages' symbol sets, is applied to all the text and its output the passed to a word unigram relative entropy model (trained on output from the same OCR system), which identifies the language.

We were led to investigate methods based on OCR output by the presumption that OCR would ultimately be performed anyway in many applications. We trained directly from OCR results derived from our training set images, rather than from extrinsic linguistic data such as corpora, for two reasons:

1. it is thus unnecessary to acquire electronic corpora for all the languages to be discriminated; and

2. the effects of OCR errors are modeled along with the language.

An additional advantage is that the training of language models from page image samples can be automated almost entirely.

[b] For an horizontal text line of upright Latin characters, the baseline is the horizontal line that touches the bottoms of characters that have no descenders.

[c] The x-height line is the horizontal line that touches the tops of characters that have no ascenders: thus it is a line along the top of a lower-case "x".

Early in the project we experimented with both character-based and word-based models. We observed that a word unigram relative entropy model gave the best accuracy, correctly identifying the dominant language in 99.5% of test images at both fine and standard resolution. Also, these models can be trained rapidly, and have excellent real-time performance.

Starting from the training and test image sets for the Latin languages (English, French, German, Italian, and Spanish), we constructed language ID training, validation, and test sets as follows:

1. Run OCR on each page image. The system we used for OCR is described in [9] and [7]. We have specialized an existing Latin-script character classifier to the ISO 8859-1 character set, which includes the 94-character printable ASCII set plus 95 other extended-Latin characters (including alphabetic characters with diacritical marks and additional punctuation forms). This character set covers the five languages used in our tests.

2. Manually edit the OCR output to remove output associated with the TSI information generated by the sending FAX machine (sending FAX ID, date and time, page numbering, etc.), so as not to bias the language models.

3. Filter the OCR output to remove lines that are very short, or which contain a large proportion of non-alphabetical characters. Filtering of the OCR output is desirable because the OCR subsystem sometimes tries to interpret ink that is not Latin text, and sometimes generates garbage when the image is of poor quality (e.g., for shaded backgrounds or colored paper).

4. Randomly select 10% of the remaining OCR output lines from training set images for each language to form a "validation" set for that language; for each language the other 90% of filtered OCR output lines constitute the training set. Validation sets are needed to adjust parameters of the training process using data that is independent of both the training and test sets.

8.1 Training Procedure

We have implemented and carried out initial tests for a word-based approach to Latin language ID. Under this approach, which we call "word unigram relative entropy," models are created jointly for the Latin languages of interest (currently English, French, German, Italian, and Spanish). In our initial tests,

we have trained models from OCR output; however, we also plan to test models trained from electronic corpora.

The training procedure works as follows:

1. Count the number of occurrences of each unique word in the training data for each language. In OCR output, a "word" is any set of characters delimited by white space. Select from each language the N most frequently occurring words (we used N=200 for our experiments). The words thus selected from OCR output may, of course, not be actual words, since particular OCR errors may occur repeatably for frequently occurring words. The union of the words selected for each language forms the basis of the language models.

2. From the union of frequently occurring words for each language, remove any words that are likely to reflect biases in the training set. The resulting set of words, plus a category "OTHER" containing all other words, is called the "model set." We chose to automatically remove any words not containing at least one alphabetical character, but made no other attempt to remove words that were subject matter specific. For N=200, and fine resolution images, the resulting merged word set contained 693 words, including subject matter specific words; this model gave excellent accuracy.

3. The model for each language is expressed by the number of occurrences of each model set word in the training set for that language, including the number of occurrences of OTHER words in the training set for that language.

Table 2 summarizes the contents of the training, validation, and test sets for this procedure.

Naturally, in each language model many of the word counts will be zero, since words that occur frequently in one language often don't occur at all in others. For example, "und" appeared only in the German training set, where it appeared frequently enough to be included in the model set. Interestingly, 60% of the model set words occurred in the training sets for more than one language, because

- They are legitimate words in more than one language.
- Words and phrases of one language appeared in the training set for another.
- OCR errors produced a word in another language.

Table 2: Test sets used for training, validation, and test of the word unigram relative entropy method for Latin language identification.

	Number of pages		Number of words					
			FINE			STANDARD		
Language	train	test	train	valid'n	test	train	valid'n	test
English	42	42	21149	2278	27589	20744	2269	27356
French	50	50	28908	3128	27257	26933	2789	25858
German	47	47	27275	2800	31485	26090	2748	31378
Italian	21	22	12796	1333	18584	10410	1091	17521
Spanish	27	27	18545	1961	20384	17013	1916	19320

Latin language ID is performed on OCR output for a page by computing the word unigram relative entropy H_L for each language L, as follows:

$$H_L = \sum_{w \in W} p_{test}(w) \log(p_{test}(w)/p^L_{model}(w))$$

where W is the model set, $p_{test}(w)$ is the relative frequency of word w in the test sample, and $p^L_{model}(w)$ is the relative frequency of w in the "model set" for language L. The language having the lowest relative entropy is selected.

8.2 Results

We trained character-level unigram, bigram, trigram, and variable-length Markov models using software provided by Dr. Isabelle Guyon, then of AT&T Bell Laboratories. The variable-length Markov models (VLMMs) were constructed and trained using techniques described in[10] (based on[11]). Results of tests on fine resolution images are shown in Table 3, in which the following terms appear:

- "fraction correct" is the fraction of 188 test images in 5 languages for which the predominant language was identified correctly;
- "number of states" is the number of states in the finite state machine associated with the language model;
- "number of parameters" is the number of parameters in the language model, equal to the number of connections in the finite state machine;
- $(H_L)_{correct}$ is the average entropy (in bits per character) for images whose language was identified correctly (lower entropies indicate better language models);

- $(\Delta H_L)_{correct}$ is the difference in average entropy per character between the best and next-best language models for each page image whose language was identified correctly; and

- $(\Delta H_L)_{error}$ is the difference in average entropy per character between the best and next-best language models for each page image whose language was identified incorrectly.

Note that the entropy figures for VLMM and word models are not directly comparable, since the VLMM entropies are in bits per character and the word model entropies are in bits per word. The average word model entropies are so low because the majority of words are in the "other word" category.

Subsequently, the VLMM and word unigram relative entropy model were tested on standard resolution FAX images as well. Surprisingly, results for standard resolution FAX images were nearly as good as for fine resolution, despite the fact that OCR accuracy was relatively poor at standard resolution. This is reflected in the lists of frequently-occurring words, which contain many words resulting from OCR errors. Apparently, OCR errors, while frequent, are predictable enough that language identification is not significantly impaired.

This, along with the work of Spitz and Ozaki[12] on Latin language ID using character shape codes, suggested we might obtain reasonably good accuracy even using a classifier not designed for the full ISO-8859 Latin 1 character set. To test this idea, we re-trained and tested the word unigram relative entropy model on OCR output generated from the same images using an ASCII (only) classifier, and found Latin language ID performance comparable to that for ISO Latin 1. Results of this are shown in Table 4.

Work reported subsequently by Sibun and Reynar[6] shows that it is possible to accurately discriminate among 27 Latin languages using N-gram statistics on character shape codes. It would be interesting to test their techniques on the low-resolution images in our database; we would expect to see somewhat lower accuracy than for our own approach, since we are taking advantage of our OCR subsystem's ability to properly segment words to characters in the many instances where multiple characters are joined.

9 System Performance

Our test set consists of 272 original pages, artificially rotated to face down, left and right to evaluate the script identification and orientation detection components. The performance of the system is summarized in Figure 5. With slight modifications to adjust for unequal resolutions in the X and Y directions,

Table 3: Results of Latin-language identification tests, using various methods, on 200x200 ppi (fine-resolution FAX) images.

	UNIGRAM	BIGRAM	TRIGRAM	VLMM	WORD
% correct	84.6	97.9	97.9	97.3	99.5
$N_{correct}$	159	184	184	183	187
N_{errors}	29	4	4	5	1
TOTAL	188	188	188	188	188
NUMBER OF STATES					
English	165	165	2171	650	1177
French	165	165	2202	587	1177
German	171	171	2908	579	1177
Italian	169	169	5014	486	1177
Spanish	168	168	2134	605	1177
NUMBER OF PARAMETERS					
English	164	27K	356K	106K	693
French	164	27K	361K	96K	693
German	170	29K	494K	98K	693
Italian	168	28K	842K	82K	693
Spanish	167	28K	356K	101K	693
$(H_L)_{correct}$ (bits/char or word): mean(std dev)					
English	5.04(.22)	4.47(.36)	4.18(.47)	4.17(.47)	0.40(.10)
French	4.97(.20)	4.30(.36)	3.98(.52)	3.95(.52)	0.46(.16)
German	5.25(.19)	4.55(.33)	4.36(.44)	4.37(.42)	0.38(.12)
Italian	5.32(.33)	4.70(.51)	4.55(.60)	4.55(.60)	0.39(.12)
Spanish	5.01(.23)	4.42(.37)	4.24(.48)	4.24(.47)	0.45(.12)
$(\Delta H_L)_{correct}$	0.12(.07)	0.44(.14)	0.71(.23)	0.70(.23)	1.03(.36)
$(\Delta H_L)_{error}$	0.07(.14)	0.31(.27)	0.43(.28)	0.27(.31)	0.10(—)

Table 4: Results of Latin-language identification tests, using the character VLMM and word unigram relative entropy methods, on fine and standard resolution FAX images. Word model results are shown for both the ISO-8859 Latin 1 character set, and for the ASCII character set.

	VLMM/Latin1		WORD/Latin1		WORD/ASCII	
	Fine	Standard	Fine	Standard	Fine	Standard
% correct	97.3	96.8	99.5	99.5	100.0	100.0
$N_{correct}$	183	182	187	187	188	188
N_{errors}	5	6	1	1	0	0
TOTAL	188	188	188	188	188	188
NUMBER OF PARAMETERS						
English	106K	113K	693	606	706	600
French	96K	102K	693	606	706	600
German	98K	106K	693	606	706	600
Italian	82K	88K	693	606	706	600
Spanish	101K	70K	693	606	706	600
$(H_L)_{correct}$: mean(std dev)						
English	4.17(.47)	5.02(.29)	0.40(.10)	0.37(.11)	0.45(.16)	0.39(.12)
French	3.95(.52)	4.81(.40)	0.46(.16)	0.43(.10)	0.50(.17)	0.48(.12)
German	4.37(.42)	5.02(.25)	0.38(.12)	0.33(.09)	0.42(.14)	0.40(.12)
Italian	4.55(.60)	5.18(.40)	0.39(.12)	0.38(.07)	0.39(.12)	0.39(.10)
Spanish	4.24(.47)	4.95(.31)	0.45(.12)	0.43(.10)	0.47(.12)	0.46(.10)
$(\Delta H_L)_{correct}$						
	0.70(.23)	0.36(.15)	1.03(.36)	0.49(.22)	1.10(.41)	0.55(.29)
$(\Delta H_L)_{error}$						
	0.27(.31)	0.05(.03)	0.10(—)	0.00(—)	N/A	N/A

the system was retrained and tested on standard resolution images. Its performance is summarized in Figure 6. The system correctly identified the language and the orientation on 97.4% of the fine resolution images, while rejecting on 2.1% of the images. On standard resolution images, the correct rate is 94.8% with 2.5% reject.

A major source of error for our system is ink associated with half tone images, line drawings, and other nontext components. Since the filter for nontext components in our current system is based on size only, many character size fragments, such as parts of graphics or photocopying artifacts, are treated as text. Features used for both script identification and orientation determination for Latin languages rely heavily on the baseline and x-height line information. Therefore, it is not surprising the performance of these two modules degraded on low resolution images. In addition, these modules can not handle pages containing only upper case alphabetical characters and numbers.

The cell density feature used for Asian languages can be sensitive to page content. A combination of light font and simple characters can mislead the system to classify a Chinese page as Japanese. Feature vectors for pages in these two languages are not well separated, suggesting higher order moments than mean and variance, or other features, are necessary to build a reliable solution.

Features used in orientation detection for Asian languages, though local and susceptible to noise in the image, are quite robust on good quality images. We are pleased that these features, originally designed only for Chinese, work on Japanese and Korean pages as well.

Although feature-based script identification and orientation detection is very efficient for the target languages, this approach is not easily extensible to other languages and scripts. Whether it is capable of handling or rejecting additional languages remains to be tested.

Of the models tested on the Latin-language identification problem, word unigram relative entropy models give the best accuracy for both fine and standard resolution images. Such models can be trained automatically from OCR output; the number of words selected from each language in forming the "model set" controls the number of parameters and the performance of the models. In addition, these models contain significantly fewer parameters than character-based models of equivalent performance, and thus will offer advantages in practical implementations.

For all these Latin-language-specific models, the number of states and parameters may be varied using an entropy-threshold criterion, so that it may be possible to tune them to obtain better accuracy/entropy figures of merit, though most likely at the expense of more memory usage and slower execution.

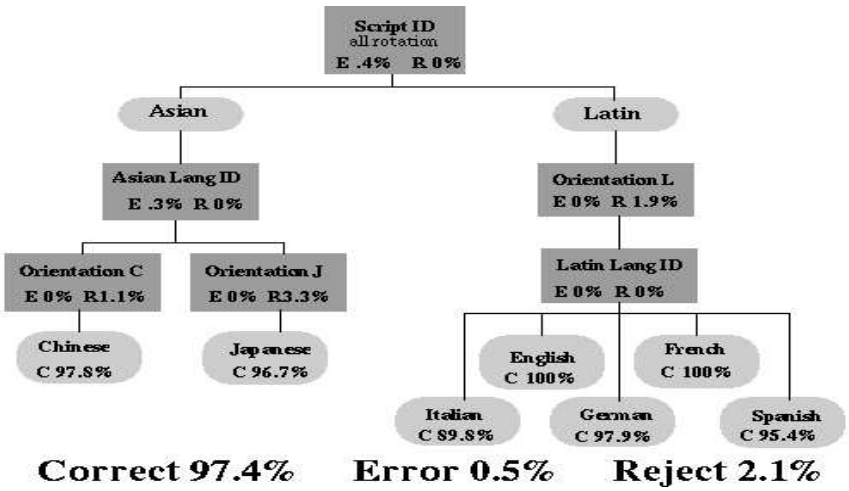

Figure 5: Summary of system performance on fine resolution pages. Accuracies of system modules, in darker gray, are also shown. Language read rates are calculated based on our database.

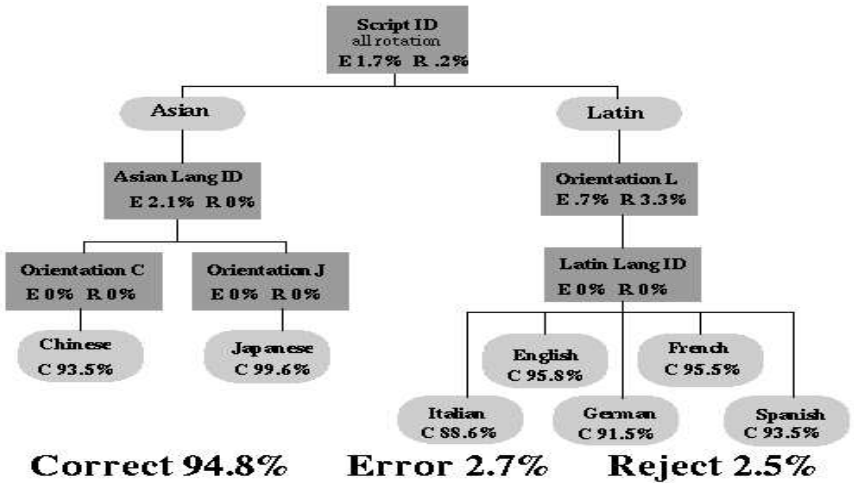

Figure 6: Summary of system performance on standard resolution images.

Using ASCII (rather than ISO-8859 Latin 1) OCR as the basis for Latin-language identification with word unigram relative entropy models caused no significant degradation in accuracy in our system, even though the classifier was presented with character images for which it was not trained.

Variable-length Markov models (VLMMs) seem to be best of the character-based Latin-language models. At the level of a full page of OCR output, character bigram, trigram, and variable-length models discriminate well among English, French, German, Italian, Spanish when trained on OCR output; character unigram models clearly do not. Based on differences in per-character entropy between best and next-best models for a given page, VLMMs should have best rejection characteristics: that is, they should give the most reliable indication of when the language is uncertain.

While our tests were conducted at the page level, it is of course possible to perform language identification at the level of blocks of text, or even text lines (and the results aggregated to higher levels if necessary). This might allow the detection of mixtures of languages within single pages. The dependence of Latin language ID accuracy on the number of words of input remains to be investigated.

One could construct a Latin-language ID system that has an additional "reject" category, intended for images where there is not enough evidence to determine the language. This may allow significantly higher accuracy for examples not classified as "reject." at the expense of having no decision on a relatively small fraction of examples. A natural criterion for rejection is the following: examples that are not rejected should have a significantly lower relative entropy per word for the best language model than for the next best (second choice) language model. For the ISO Latin 1 word unigram models, requiring an entropy difference of at least 0.1 bits per word between top choice and second choice models causes only one example to be rejected — the single image for which the top choice language is incorrect! This is significantly better rejection performance than is achieved by the character-level models.

Our reliance on classification (OCR) for discriminating among Latin-script languages had the advantage of economy of means (since OCR facilities already existed), but we may have paid a penalty in speed that may not be required merely to achieve good language identification.

The running time of the system naturally depends on the complexity of the page. Preprocessing and layout analysis consume from 1 second (on a typical business letter) to 4 seconds (on a technical journal page), running on a Sparc20-class machine. Extraction of various features for script identification, Asian language identification and orientation detection takes between 1 and 5 seconds. Character recognition, required for Latin language identification,

is performed by a neural network classifier at 80 recognitions per second, and requires from 30 to 90 seconds a page on our database. Currently, the system is implemented as several separate programs interfaced by reading and writing to external files. More efficient processing can be achieved by tighter integration of these programs and code optimization.

10 Conclusions and Future Work

We have extended script and language identification to handle cases significantly more challenging than previously reported, combining the effects of complex layouts, unoriented pages, and degraded images. Tests on FAX images show that our method accurately identifies scripts (98.16%), language and page orientations (94.76%) at standard (low) resolution. We hope that our method will, with few essential changes, extend to a much wider range of scripts and languages, particularly other European non-Latin scripts such as Greek and Cyrillic and connected scripts such as Devanagari (Brahmi) and Arabic.

The dependency of Latin language identification accuracy on the number of languages modeled is currently unknown and should be investigated.

Language models could be trained directly from electronic corpora; the performance of such models for Latin language identification is unknown. It is expected that such models would work well when OCR accuracy is good, or when OCR errors can be effectively modeled.

The rapid success of so many disparate attacks on these problems, under progressively more challenging conditions, has surprised us somewhat, and we speculate that significantly faster, more accurate, and more versatile methods may remain to be discovered.

In the document image analysis research community, until recently almost all methods reported for the recognition of machine-printed text were restricted to a single language. In the last five years a few researchers have described "multilingual" text recognition technology. One of these is a generic page reader [9] capable of being easily retargeted to (reengineered for) new languages. The resulting semi-automatically engineered page readers are restricted to reading *a single known* language at a time. Another system [12] has shown promising early success in using language identification to switch, on the fly, between several pre-existing language-specific page readers (using disparate technology). Now we may be within sight of the ability to read two or more languages at a time in a very strong sense, that is, as they are encountered without warning in different parts of a document or within an unconstrained stream of documents, using a single generic page-reader technology.

Acknowledgements

Our approach to this problem was influenced decisively from the beginning by the generously shared experience of Larry Spitz of Daimler-Benz Research. We owe Larry a second debt of gratitude for his cooperation as we assembled the image data base, which will allow us systematically to compare our results with his in future experiments. Dr. Judith Hochberg of Los Alamos National Laboratory (LANL) kindly provided us with 268 images of machine printed pages in 13 scripts which we hope to use similarly. Nikos Annitsakis assisted us in the collection of our image database, and other members of Bell Labs, too numerous to name, responded to our appeal for multilingual documents. The language-free layout software [7] on which we have depended was largely the creation of David J. Ittner, in collaboration with the third author. We are grateful to Dr. Isabelle Guyon for discussions of VLMMs and for her software for training these on character-level models. We have benefited from stimulating discussions with Tin Kam Ho and John Hobby.

References

1. A. L. Spitz, Script and Language Determination from Document Images, *Proc., 3rd Symp. on Document Anal. and Info. Retrieval*, pp. 229–235, Las Vegas, Nevada, 1994.
2. P. Sibun and A. L. Spitz, Language Determination: Natural Language Processing from Scanned Document Images, *Proc., Applied Natural Language Processing*, pp. 115–121, Stuttgart, 1994.
3. T. Nakayama and A. L. Spitz, European Language Determination from Image, *Proc., Int'l Conf. on Document Anal. and Recog.*, pp. 159–162, Tsukuba, Japan, 1993.
4. J. Hochberg, L. Kerns, P. Kelly, and T. Thomas, Automatic Script Identification from Images using Cluster-based Templates, *Proc., 3rd Int'l Conf. Document Anal. and Recogition.*, pp. 378–381, Montreal, Canada, August 14–16, 1995.
5. S. L. Wood, X. Yao, K. Krishnamurthi, and L. Dang, Language Identification for Printed Text Independent of Segmentation, *Proc., Int'l Conf. on Image Processing*, October 23–25, 1995.
6. P. Sibun and J. C. Reynar, Language Identification: Examining the Issues, *Proc., 5th Symp. on Document Anal. and Inf. Retrieval*, pp. 125–135, Las Vegas, Nevada, April 15–17, 1996.
7. D. J. Ittner and H. S. Baird, Language-Free Layout Analysis, *Proc., IAPR 2nd Int'l Conf. on Document Analysis & Recognition*, pp. 336–

340, Tsukuba Science City, Japan, October, 1993.
8. H. S. Baird, D. Gilbert, D. J. Ittner, A Family of European Page Readers, *Proc., IAPR 12th Int'l Conf. on Pattern Recognition*, Jerusalem, Israel, October 9–13, 1994.
9. H. S. Baird, Anatomy of a Versatile Page Reader, *Proc. of the IEEE*, Special Issue on OCR, vol. 80, no. 7, pp. 1059–1065, July, 1992.
10. I. Guyon and F. Pereira, Design of a Linguistic Postprocessor using Variable Memory Length Markov Models, *Proc., 3rd Int'l Conf. Document Anal. and Recogition*, pp. 454–457, Montreal, Canada, August 14–16, 1995.
11. D. Ron, S. Singer, and N. Tishby, The Power of Amnesia, in J. Cowan *et al.* (Eds), *Advances in Neural Information Processing Systems*, vol. 6, Morgan Kauffmann, 1994.
12. A. L. Spitz and M. Ozaki, Palace: a Multilingual Document Recognition System, in A. L. Spitz and A. Dengel (Eds.), *Document Analysis Systems*, World Scientific, Singapore, 1995.