Software Tools and Test Data for Research and Testing of Page-Reading OCR Systems

Thomas A. Nartker, Stephen V. Rice, and Steven E. Lumos
Information Science Research Institute (ISRI)
University of Nevada, Las Vegas
Las Vegas, NV 89154-4021, USA
702-895-3338, FAX 702-895-1183
Tom@isri.unlv.edu, Rice@cs.olemiss.edu, SLumos@isri.unlv.edu

ABSTRACT

We announce the availability of the UNLV/ISRI Analytic Tools for OCR Evaluation together with a large and diverse collection of scanned document images with the associated ground-truth text. This combination of tools and test data will allow anyone to conduct a meaningful test comparing the performance of competing page-reading algorithms. The value of this collection of software tools and test data is enhanced by knowledge of the past performance of several systems using exactly these tools and this data. These performance comparisons were published in previous ISRI Test Reports and are also provided. Another value is that the tools can be used to test the character accuracy of any page-reading OCR system for any language included in the Unicode standard. The paper concludes with a summary of the programs, test data, and documentation that is available and gives the URL where they can be located.

Keywords: Optical Character Recognition, Software Tools, Testing

1. BACKGROUND

The curse of many "modern" technologies is that comparing competing systems is beyond the capacity of all but the largest organizations. Both individual and corporate users are at the mercy of inconclusive reports typically found in a trade magazine. Even large corporations do not have the resources needed to evaluate the performance of page-reading OCR systems for their specific applications. At the same time, an individual researcher has no hope of identifying problems and developing algorithms that can further the state-of-the-art.

There are two dimensions to this problem. One especially important consideration is the existence of performance metrics (and programs that implement them) that measure important attributes of systems to be considered. For page-reading systems, the standard performance measure is "character accuracy." Tools to test page-reading systems must however, measure many other aspects of performance that may be important for some applications. In addition to character accuracy, one might wish to measure "accuracy by character class," "non-stopword accuracy," "marked character efficiency," or "automatic zoning." The performance measurement tools we present are those developed at ISRI as part of a program to test and publicize the state of page-reading technologies that was conducted from 1992 until 1996.

A second problem is the cost of preparing a significant quantity of test data. The cost of manually retyping printed pages and insuring very high accuracy is usually prohibitive. Also, it is not clear how example test pages should be selected. As part of ISRI's test program, we established the practice of defining test samples from different document domains, each containing 200-400 pages (several hundred thousand characters) selected at random.

Even when test tools and data are available, the task of extracting meaningful results from test output can be difficult. The ISRI annual tests from 1993 through 1996 show an example of the complexity of presenting and understanding the results of such tests.

Finally, we also offer some insight into problems facing contemporary OCR systems. The goal of ISRI's OCR test program in the 1990's was not only to publicize the state-of-the-art of page-reading systems but to provide information that would make possible improvements in this technology. We learned that it was easy to produce endless lists of different confusions (errors) made by such systems. Identifying OCR errors that deserved special attention was more difficult.

The result of this effort was the publication, in 1999, of a research monograph containing a set of problems "at the frontier" of OCR [1]. This book, subtitled "An Illustrated Guide to the Frontier," presented 280 common OCR problems that were easily solved by most grade school children, and that appear solvable by computer if specific focus were applied. Shortly after the publication of this book, ISRI made available a CD containing the 280 images, the associated ground-truth text, and executable copies of the character and word accuracy programs described above. This distribution, called "The OCR Frontiers Toolkit," was meant to be a resource for graduate students searching for an opportunity to solve some of the lingering problems in OCR. This toolkit is also provided on the Web site.

In the following Sections, we give an overview of the set of measurement tools, a description of the ground-truth data samples, an overview of the ISRI test reports, a description of the Frontiers toolkit, and the URL where they can be found. We refer to this collection as the ISRI OCR performance toolkit.

2. ANALYTIC TOOLS

The performance measurement tools provided are shown in Table 1. Nineteen programs are provided for testing in four different categories. The categories of testing supported are shown in the first column.

Testing	Program	General Functionality
Character Accuracy		
	accuracy	Compute character accuracy statistics for one page of text
	accsum	Compute character accuracy statistics for multiple pages of text
	synctext	Synchronize the characters of two or more text strings
	groupacc	Compute the accuracy of groups of characters
	accci	Compute 95% confidence interval for a set of character accuracy tests
	accdist	Compute the distribution of accuracies found in a set of reports
	ngram	Compute n-gram statistics for one or more text files
	vote	Compute an output string by majority voting of three or more inputs
Word Accuracy		
	wordacc	Compute word accuracy statistics for one page of text
	wordaccsum	Compute word accuracy statistics for multiple pages of text
	nonstopacc	Compute accuracy as a function of the number of non-stopwords
	wordaccci	Compute 95% confidence interval for a set of word accuracy tests
	wordaccdist	Compute the distribution of word accuracies found in a set of reports
	wordfreq	Computes the frequency of words in one or more text files
Automatic Zoning		
	editop	Estimate the edit operations needed to correct one page of OCR text
	editopsum	Compute the edit operations needed to correct a set of pages
	editopcost	Compute the cost of edit operations needed to correct a set of pages
Foreign-Language OCR		
	asc2uni	Convert an Extended ASCII file to a Unicode file
	uni2asc	Convert a Unicode file to an Extended ASCII file

Table 1. ISRI Analytic Tools for OCR Testing

All programs are Unix shell programs that are suitable for use in batch-oriented shell scripts. Each program takes command-line parameters and performs its task in a non-interactive manner. The programs are written in C and should compile without change on most systems. In general, the main programs compare the OCR-generated file for a page with the correct (or "ground truth") file for that page and compute measures of OCR performance.

2.1. Character Accuracy Related Programs

2.1.1. Character Accuracy Reports

The two main programs provided are **accuracy** and **accsum**. **accuracy** computes "character accuracy" for a page and data for several other character-related metrics. These include the accuracy improvement achievable if all "marked" characters were corrected, the accuracy of uppercase, lowercase, digit, space, and special characters, as well as a list of all confusions and the accuracy of each different character on the page. This file of character accuracy metrics is called an *accuracy_report*.

Because significant performance tests are seldom produced from a single test page, the program **accsum** can be used to combine two or more *accuracy_reports* to produce an aggregate *accuracy_report*. Thus, **accsum** will compute the character accuracy metrics for a set of pages.

The only difference between the output of the **accuracy** program and the **accsum** program is that the percentages of characters correctly recognized are from a set of pages instead of from a single page. Since there are usually many more individual characters on a set of pages than on a single page, **accsum** outputs can be very long.

2.1.2. Character Accuracy Post-Processing Programs

For any group of pages tested there is a need to estimate the significance of the accuracies measured. The **accci** program will compute an approximate 95% confidence interval, using the method of jackknife estimation, for character accuracy from a set of *accuracy_reports*.

Two other programs can be used for specific performance measures. **groupacc** can be used to measure accuracy by character class. **groupacc** takes a set of characters, called a *groupfile*, and an *accuracy_report* and summarizes the character accuracy data from the *accuracy_report* for the characters in the *groupfile*. A common use of **groupacc** is to compute the accuracy of uppercase letters, lowercase letters, digits, etc. separately. Finally, the **accdist** program computes the distribution of accuracies found in a set of *accuracy_reports*.

2.2. Word Accuracy Related Programs

2.2.1. Word Accuracy Reports

The two main word accuracy programs provided are **wordacc** and **wordaccsum**. The **wordacc** program computes the percentage of correctly recognized words on a page and several other word-related metrics. Words are defined to be any sequence of one or more letters. Because information retrieval systems, when searching for specific words, are normally not sensitive to case, the word accuracy measured is case insensitive.

Additional word-related metrics produced by **wordacc** include a listing of the percentage of correct stopwords[*1] and non-stopwords tabulated by word-length. Also, the percentage of correct distinct non-stopwords tabulated by frequency of occurrence and the percentage of correct phrases tabulated by phrase length are computed for the page. Finally, the percentage of correct occurrences of each distinct stopword and non-stopword are computed for the page. This file of word accuracy metrics is called a *wordacc_report*.

Because significant performance tests are seldom produced from a single test page, the program **wordaccsum** can be used to combine two or more *wordacc_reports* to produce an aggregate *wordacc_report*. Thus, **wordaccsum** will compute the word accuracy metrics for a set of pages. Since there are usually many more individual words on a set of pages than on a single page, ***wordaccsum*** outputs can be very long.

*1 An ordered list of the stopwords to be used is provided as an input file to the **wordacc** program.

2.2.2. Word Accuracy Post-Processing Programs

For any group of pages tested there is a need to estimate the significance of the word accuracies measured. The **wordaccci** program will compute an approximate 95% confidence interval, using the method of jackknife estimation, for word accuracy from a set of *wordacc_reports.*

Another word accuracy post-processing program is **nonstopacc**. This program accepts an ordered list of N stopwords in *stopwordfile* and a *wordacc_report* and produces the non-stopword accuracy of this report for each subset of stopwords beginning with 0 through all N stopwords. These x y pairs are the data needed to plot the non-stopword accuracy curve presented in ISRI's 5th annual test of OCR accuracy. Finally, the **wordaccdist** program computes the distribution of word accuracies found in a set of *wordacc_reports*.

2.3. Automatic Zoning Programs

When a page-reading OCR system is operated in "automatic zoning" mode, it attempts to locate all text regions on the page and to determine their correct reading order. The system finds columns of text, and if they are not part of a table, defines a separate zone for each column so that the generated text will be de-columnized. The special problem here is not to decolumnize tables of text.

Since OCR devices attempt to locate all text on the page, an automatic zoning metric can be based on the number of character string (or text line) moves required to transform the OCR output to the correct reading order. In 1993, ISRI staff developed a zoning metric based on this number of moves called the "cost of automatic zoning" [3]. The three programs identified as "automatic zoning" can be used to compute this metric for a page of OCR output.

editop compares the correct text file for a page with the OCR-generated file and produces an *editop_report* for the page. It shows a list of the number of string moves needed tabulated by the number of the characters in the move. The **editopsum** program combines two or more *editop_report's* to produce an aggregate *editop_report*. As was true of **accsum** and **wordaccsum***,* the length of **editopsum** outputs can become large.

editopcost computes the cost of the edit operations described in an *editop_report*. If two *editop_report's* are provided, **editopcost** computes the cost of the edit operations in the first *editop_report* minus the cost of operations for the second *editop_report*.

Thus, the "cost of automatic zoning" is computed by providing two *editop_reports*. The first is an aggregate report that indicates the edit operations for correcting OCR-generated text that was produced with automatic zoning. The cost of these operations is the total cost of correcting both automatic zoning errors and character recognition errors. The second *editop_report* is an aggregate report produced for the same set of pages using manually-defined zones. The cost of these operations is the cost of correcting only the character recognition errors. Subtracting the cost of the *editop_report* for manual zoning from the cost of the *editop_report* for automatic zoning yields the cost of correcting the automatic zoning errors.

The "cost" computed by *editopcost* is based on the number of insertions, the number and lengths of move operations, and a threshold value used to convert move operations into an equivalent number of insertions. Thus, the **editopcost** output is a tabulation of the resulting number of insertions computed for each threshold value from 0 to 100 and the "cost" is displayed as a curve plotted from this data. Lower curves on a graph indicate superior automatic zoning performance. Graphs showing the result of a test comparing the cost of automatic zoning are shown in ISRI's 1994, 1995, & 1996 annual reports.

2.4. Foreign Language Testing Programs

In Sections 2.1 through 2.3, we discussed the ISRI Analytic tools assuming all page-reading systems tested were English language recognizers generating 8-bit ASCII files and all ground-truth files contained English text in 8-bit ASCII format. In fact, simply by allowing OCR output and ground-truth files to be presented in Latin1, these same routines can be used to test OCR accuracy from page readers for 13 additional languages. By allowing OCR output and ground-truth files be presented in an "Extended ASCII" format, these routines can be used to test OCR accuracy from page readers for any Unicode language.

2.4.1. Latin1 LANGUAGES

Latin1 is the ISO 8859-1 standard 8-bit character encoding. This encoding includes the 7-bit ASCII standard as a subset as well as characters for the Danish, Dutch, Faroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish languages. Although ISRI has performed tests of Spanish and German page readers, all of the character and word accuracy programs described in Sections 2.1 and 2.2 should perform properly given any Latin1 ground-truth file and output from a corresponding Latin1 page-reader. Examples are shown in the ISRI 1995 and 1996 test reports.

2.4.2. Unicode LANGUAGES

The Unicode standard specifies a 16-bit character encoding for nearly all of the world's languages. It includes Latin1 as a subset. Each of the programs described above can be used to test Unicode page readers using "Extended ASCII" as an intermediate representation.

This "extended" form of testing works as follows. First, a Unicode editor can be used to enter Unicode characters. Two routines provided in this collection, **uni2asc** and **asc2uni**, can be used to convert text files to or from Extended ASCII. Finally, all of the accuracy programs described will accept Extended ASCII files as input and produce Extended ASCII files as output. Using this scheme, the tools described above can be used to perform OCR testing for any Unicode page reader.

Although there is no reason a user needs to know, the definition of "Extended ASCII" is as follows. **uni2asc** copies all Latin1 characters (as 8 bit characters) directly to the output text file. Other Unicode characters are copied as four hexadecimal digits surrounded by angle brackets (e.g., <03A9> is the capital Omega character). **asc2uni** does the opposite to restore a Unicode file. Although Extended ASCII files can be printed, it is usually preferable to restore Unicode files and to use the Unicode editor to print the source language characters.

There is, of course, one restriction on testing Unicode languages. Because we have defined a "word" to be any sequence of one or more letters, the word accuracy programs described may not be applicable. For languages that compose words using other symbols, word accuracy is not available.

2.5. Other Programs

Finally, there are several general-purpose programs. The **synctext** program can be used to show the alignment of two or more text files. Three different synchronization algorithms are provided, each for a slightly different synchronization task. **synctext** provides an external interface to all three algorithms for general-purpose use.

If exactly two input files are specified (one correct file and one OCR-generated file), the alignment computed by **synctext** is the same as the one computed by the **accuracy** program. This allows the user to see where OCR errors occurred within the text. The algorithm used to compute this alignment is described in reference [2]. A second alignment algorithm can find "transposed" matches between two input files (see reference [3]). This algorithm is used by the **editop** program for automatic zoning evaluation. Finally, the third is an algorithm used by the **vote** program to align three or more OCR output strings. [4]

Two programs are provided to compute text file statistics regarding characters or words. **ngram** computes n-gram statistics for one or more text files and **wordfreq** computes the frequency of words in one or more text files.

The **vote** program was developed by ISRI staff to experiment with majority voting of three or more OCR systems. In the early 1990's, several authors reported significant improvements in character accuracy using some form of majority voting of several different systems. This program first aligns the input files so that agreements and disagreements are evident and a majority vote is taken to resolve disagreements. Heuristics are applied to break ties.

Although the algorithms used by some of these routines are also employed by the programs described above, **synctext**, **ngram**, **wordfreq**, and **vote** are included for general use and are not explicitly needed when doing character and word accuracy tests.

2.6 Documentation

More detail, including references describing these metrics, is provided by a user manual titled: The ISRI Analytic Tools for OCR Evaluation. Although each program is accompanied by *man* pages, the most thorough documentation can be found in Steve Rice's Ph. D. Dissertation, which is also provided. The "Frontiers Toolkit" (see Section 5 below), is recommended for beginning users of these programs.

2.7 Source Files Available

In addition to one C source file for each of the 19 programs described above, there is a library of 15 supporting modules (see Table 2). There is a ".c" and a ".h" file for each module. A list, and brief description, of these modules is shown below.

These routines, together with a general-purpose graphing program, enable researchers to display the results of a set of performance experiments in graphical form similar to that shown in the series of ISRI annual reports provided (see Section 4).

Module	Description
accrpt	Reading and writing character accuracy reports
charclass	Determining character classes
ci	Computing confidence intervals
dist	Computing the distribution of accuracies
edorpt	Reading and writing edit operation reports
list	Linked list operations
sort	Sorting
stopword	Identifying stopwords
sync	Aligning strings
table	Hash table operations
text	Reading and writing text files
unicode	Reading and writing Unicode files
util	Utility routines
wacrpt	Reading and writing word accuracy reports
word	Parsing to extract words

Table 2. Supporting C modules

3. TEST DATA

Nine different test datasets, shown in Table 3 below, are provided. These nine were used in one or more of ISRI's annual tests between 1993 and 1996. The table shows the number of pages, the number of characters in each sample, and the years it was used in the ISRI test.

Because image quality is a major factor controlling recognition accuracy, ISRI staff took the time to create several different images of each page for most test samples. For eight of the samples, each page was scanned (manually) four times producing 200, 300, and 400 dpi binary images and a 300 dpi greyscale image. For two samples, both standard mode and fine-mode FAX images were also collected. Care was taken to insure that the same ground-truth characters were appropriate for corresponding zones of each image.

For each test of recognition performance, we attempted to show test results for several different test samples to show the performance variation that was typical for different document domains. Also, each year we chose to conduct tests on at least one sample that had been used the previous year. By comparing such year to year tests, it was possible to measure performance improvements in some of the systems.

Ground-truth Test Datasets		Number of		Image Resolution						Used in Annual Test
Sample Name	Description	Pages	Characters	200 dpi bin	300 dpi bin	400 dpi bin	300 dpi grey	Fine-mode Fax	Std-mode Fax	
Sample 2	DOE Sample 2	460	817,946		x					1993 & 94
Sample M	Magazine Sample	200	666,134	x	x	x	x			1994 & 95
Sample N	Newspaper Sample	200	492,080	x	x	x	x			1995
Sample B	Business Letter Sample	200	319,756	x	x	x	x	x	x	1995 & 96
Sample L	Legal Document Sample	300	372,098	x	x	x	x	x	x	1996
Sample S	Spanish Newspaper Sample	144	348,091	x	x	x	x			1995 & 96
Sample 3	DOE Sample 3	785	1,463,512	x	x	x	x			1995 & 96
Sample R	Annual Report Sample	300	892,266	x	x	x	x			1996
Sample Z	Magazine Sample 2	300	1,244,171	x	x	x	x			1996
Totals		**2889**	**6,616,054**							

Table 3. List of Test Datasets Available

3.1 Sample Selection

Because we had access to large numbers of scientific and technical documents from the U.S. Department of Energy, we selected pages at random from this collection. These documents had been published over a 30-year period and covered a wide variety of technical subjects. In addition, although all pages were readable, many documents were photocopies and the page and print quality varied significantly for this collection.

We created a business letter and a legal document sample, because each represented a different document domain where OCR technology would frequently be needed. Although page and print quality were generally higher for these collections, business letters employed a wide variety of typefaces and type styles while the legal documents were much more uniform.

We expected that test samples containing pages from recently printed U. S. magazines would present yet another document domain but would provide higher quality images. The corporate annual reports sample was chosen because it would present very high quality images that would enable OCR systems to demonstrate their best performance. At the other extreme, the newspaper samples would generally produce lower quality images and would be a greater challenge to most OCR engines.

3.2 Files Provided

All images are provided as compressed TIFF images. Bi-tonal images are compressed with the CCITT group 4 scheme and greyscale images with LZW. FAX images are compressed with the CCITT group 3 scheme. For each sample page, there is an ASCII file containing an ordered list of zoning coordinates for the text regions on the page. For each zone of each page, there is an ASCII file containing the ground-truth text for the zone. The naming conventions for these files is described in a readme file associated with the collection.

During the last eight years, we have discovered one mistake in these test samples. In the "Annual Report Sample," the scanned image of one page is corrupt. Because this error was discovered after the results for the 5th annual test were published, we have not replaced the image in the ground-truth with a corrected image. Thus, the error associated with this corrupt image is reflected in the 1996 test results and still exists in the test data.

3.3 Conventions for OCR Output and Ground-truth Text

Typically, OCR systems generate some output character, called a reject character, when the system is unable to determine what character corresponds to the dots on the image. The accuracy program described above assumes that a "~" is used as the reject character. OCR systems also generate "suspect" characters to mark an output character that is of low confidence. The accuracy program assumes that the "^" character is used as a prefix to mark the suspected character.

Because both horizontal and vertical spacing is compressed by accuracy prior to matching with OCR output, spacing can be inserted in the ground-truth files to retain readability. Blank lines, and leading and trailing blanks on a line, are eliminated. Consecutive blanks within a line are compressed to a single blank.

Non-ASCII symbols appearing on a page, such as a bullet symbol or a Greek letter, require special handling. Each non-ASCII symbol is represented in the correct text by a tilde (~). Since OCR devices are not expected to recognize these symbols, the error-counting software allows zero or one arbitrary characters to be generated for each tilde without charging an error.

4. ISRI ANNUAL REPORTS

These reports show a five-year progression of attempts to do a more thorough job in presenting a picture of exactly what performance could be expected of contemporary OCR systems. There was some attempt to compare products to stimulate competition and to advance the technology, but the overall goal was to establish what performance range was provided by contemporary systems and to identify where improvements were most needed.

4.1 Performance Ranges Observed

First, although standard-mode FAX images produced the lowest accuracy OCR output, we found the influence of image quality in other document domains to be less dominant than we had expected. Most obvious was that the use of imaginative typography, especially colored print, colored backgrounds, and reverse video, contributed to lower accuracy results for both corporate annual reports and U. S. Magazines.

The average accuracy of newsprint samples, on the other hand, was as good or better than that of corporate annual reports. The absence of such typographic variability and the uniformity of typeface and type styles resulted in uniformly high accuracies for our legal document sample.

In general, we found the measured average accuracies of our test samples to exhibit more variability between devices and between samples than we expected. Overall, this series of tests did serve to highlight the range of performance that could be expected, but dominant trends were hard to find.

Above 99% character accuracy was not common although, after correcting all marked characters, was frequently achievable. Character accuracy greater than 99.6% almost never occurred even after correcting marked characters. Character accuracy below 96% was not common for these test samples, with most systems producing between 97% and 99% on most samples. Greyscale OCR systems frequently outperformed their binary counterparts in character accuracy by ½ to 1%. The average accuracy of both uppercase characters and digits was frequently two or more percentage points lower than for lowercase characters. The average character accuracy resulting from 300 dpi images was frequently one to two percent higher than for 200 dpi images. Images scanned at 400 dpi, however, normally produced no increase in average accuracy. In fact, lower accuracy at 400dpi than at 300dpi was more common than one would expect. For most test samples, page quality seemed to be an issue in that 20% of the pages contributed about 80% of the character errors.

The variation in non-stopword accuracy for different test sets was greater than expected. For some datasets all systems produced a non-stopword accuracy below 95% and for other datasets an accuracy above 98%. It was not uncommon for there to be three or more percentage points difference in non-stopword accuracy between the best and worst system in a test. Also, there was a wide variation in the performance of different systems in automatic zoning. Factors of two to three times more cost were common with no one system consistently best for all test samples.

While this series of tests shows how difficult understanding test output can be, it also illustrates that the difference in performance between two systems can easily vary by factors of two to five in residual errors.

4.2 Where Improvements are Most Needed

We believe that the improvement most needed in current OCR technology is in non-stopword accuracy. An important application of OCR systems is in recovery of information from large archival collections of printed documents. These projects invariably make use of information retrieval (IR) technologies to enable searchers to find information. All IR systems utilize only non-stopwords in constructing the document index which is used to automate the search. Improved accuracy of stopwords, of digits, and of punctuation is of little concern in retrieval.

Improvements are also needed in the performance of automatic zoning. Like non-stopword accuracy, automatic zoning is important in archival conversion.

In areas more important for desktop use, marked character efficiency needs to be improved. Regardless how many errors are made by an OCR system, performance can be quite satisfactory if they are easily found. Less important, but also obvious from experimental results, improvements are needed in uppercase and numeric digit recognition.

In our final effort to identify improvements needed, we produced a collection of OCR problems derived from "likely sources of error." This collection of OCR problems is described in Section 5 below.

5. THE FRONTIERS TOOLKIT

After spending several years attempting to expose the state of the art of page recognition technologies, we turned our attention to what could be done to improve them. Having tested many different systems using different test datasets over several years, we certainly had access to an enormous supply of errors that contemporary devices make. Although it would not be an easy job, we speculated that we might be able to identify some subset of these that, if corrected, could produce an improvement in the technology.

We began by utilizing three leading OCR systems to process pages from four types of documents: technical reports, magazine articles, newspaper articles, and business letters. Using a scanner, each page was digitized at a resolution of 300 dots per inch to produce a binary image, which was submitted to each system for conversion. To locate errors, OCR output was compared with correct text using the **accuracy** program.

We randomly selected about 1000 words and phrases in which more than one system made an error and then analyzed the error snippets to determine the likely sources of error. We organized these sources into a taxonomy that ultimately provided the structure for the "Frontiers" book. Table 4 below shows four Chapter headings that identify the main categories of error we identified. Each Chapter is further divided into subcategories showing a slightly different example of the error. The right hand column of Table 4 shows the number of different examples we found of each subcategory.

Each example was a small section (i.e., a snippet) of a page image with between one and four words. Also associated with each snippet is the correct text and the text produced by each of the three OCR systems. In each case, the correct characters are easily identified by a human reader. In each case, some characteristic of the snippet (e.g., heavy print) has made a strong contribution to the error.

We believe this collection of problems provides a challenge for graduate students to produce recognition algorithms that will solve these problems and ultimately, improve page-reading technologies.

The Frontiers Toolkit contains a user manual, the test data, the character and word accuracy test tools, sample output for the tools from three different OCR systems, and sample character accuracy and word accuracy reports for each OCR system. It is organized to make it as easy as possible for graduate students to begin using the toolkit by conducting character accuracy and word accuracy tests that can be verified by comparing with the correct output provided.

The Toolkit user manual is a subset of the Analytic Tools user manual corresponding to the character and word accuracy tools only. The test data includes not only the images (and correct text) for each of the 280 snippets, but also the images (and correct text) for the entire page containing each snippet. For each page there is a zone file and for each zone defined there is a separate correct text file. The naming conventions for these files is described in a readme file.

Chapter	Chapter and Section Number	Sub-Category	Number of Snippets
Imaging Defects	2.1	Heavy Print	35
	2.2	Light Print	35
	2.3	Heavy and Light Print	8
	2.4	Stray Marks	17
	2.5	Curved Baselines	6
Similar Symbols	3.1	Similar Vertical Symbols	22
	3.2	Other Similar Symbols	25
Punctuation	4.1	Commas and Periods	25
	4.2	Quotation Marks	12
	4.3	Special Symbols	10
Typography	5.1	Italics and Spacing	17
	5.2	Underlining	8
	5.3	Shaded Backgrounds	8
	5.4	Reverse Video	13
	5.5	Unusual Typefaces	9
	5.6	Very Large Print	15
	5.7	Very Small Print	15
Total			280

Table 4. OCR Problems from the Frontier

The character accuracy and word accuracy programs are provided as binary executables. In fact, there are four different executable sets, one for the Irix environment, one for Linux, one for Solaris, and one for Windows. These are also described in the readme file.

Finally, there are sample character and word accuracy reports for each snippet page for each of the three OCR systems. A description of these files and their naming conventions is provided in the readme file.

AVAILABLE ON THE WEB SITE

URL: <http://www.isri.unlv.edu/OCRtk/>

Test Tools:

1. The ISRI Analytic Tools for OCR Evaluation: User Manual (ISRI TR-96-02, August 1996)
2. The ISRI Analytic Tools for OCR Evaluation: Source Code (C)
3. Measuring the Accuracy of Page Reading Systems (Ph D Dissertation, S. V. Rice, December 1996)

Test Data:

4. Test Datasets

ISRI Annual Tests of OCR Accuracy:

5. 1993 Annual Test
6. 1994 Annual Test
7. 1995 Annual Test
8. 1996 Annual Test

OCR Frontiers Toolkit:

9. The OCR Frontiers Toolkit: User Manual
10. OCR Frontiers test data
11. Character accuracy and word accuracy test tools (four different executable versions)
12. Sample output from three contemporary OCR systems
13. Sample character and word accuracy reports for each OCR system

REFERENCES

[1] S. V. Rice, G. Nagy, and T. A. Nartker, *OPTICAL CHARACTER RECOGNITION: An Illustrated Guide to the Frontier*, Kluwer Academic Publishers, Norwell Massachusetts, 1999, ISBN 0-7923-8492-X.

[2] S. V. Rice, *Measuring the Accuracy of Page-Reading Systems*. Ph.D. dissertation, UNLV, Las Vegas, 1996. (see Test Tools 3 above)

[3] J. Kanai, S. V. Rice, T. A. Nartker, and G. Nagy. Automated Evaluation of OCR Zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):86-90, 1995.

[4] S. V. Rice, J. Kanai, and T. A. Nartker. An algorithm for matching OCR-generated text strings. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1259-1268, 1994.