The ISRI Analytic Tools

for OCR Evaluation

Version 5.1

Stephen V. Rice and Thomas A. Nartker

Information Science Research Institute

TR-96-02

August 1996

Contents

1 Introduction

The Information Science Research Institute (ISRI) at the University of Nevada, Las Vegas (UNLV) tested optical character recognition (OCR) systems on an annual basis from 1992 through 1996. The OCR systems that were evaluated are known as "page readers." These systems take as input a bitmapped image of any document page and attempt to locate and identify the machine-printed characters on the page. Each annual test is described in a technical report (see references [1] through [5]). Synopses of the test results have been published in *Inform* magazine (references [6] through [9]).

Since 1991, ISRI has been actively developing performance measures for page-reading systems. These measures have been used in the annual test and enable a comprehensive evaluation of these systems. The measures include character accuracy, marked character efficiency, word accuracy, non-stopword accuracy, phrase accuracy, and the cost of correcting automatic zoning errors. With the exception of the automatic zoning measure, which is described in [10], a formal specification of these measures, including algorithms for computing them, is presented in Steve Rice's doctoral dissertation [11].

The ISRI OCR Experimental Environment is a suite of software tools for conducting large-scale, automated tests of page readers. The ISRI Analytic Tools is a subset of this suite and includes the programs that compute the performance measures. These programs are described in this document one by one with examples.

1.1 Operating Environment

The Analytic Tools are Unix shell programs that are suitable for use in batch-oriented shell scripts (which are essential to large-scale, automated testing). Each program is given command-line parameters and performs its task in a non-interactive manner. If processing is successful, a zero exit status is returned; otherwise, a non-zero status is returned and an error message is written to "stderr," the Unix standard error file.

The usage of a program can be displayed by entering the name of the program with no arguments. Specifying "–h" or "–help" as an argument will also display the usage. A *man* page is available for each program.

The programs were developed for Sun SPARCstations under SunOS 4.1.3. They should operate under Solaris without modification. The programs are written in C and could be ported to other platforms.

1.2 Special Characters

The user must supply OCR-generated text files and correct (“ground truth”) text files. The Analytic Tools include programs to compare an OCR-generated file with the correct file and obtain measures of performance.

A tilde (~) in an OCR-generated text file is treated as a reject character. A circumflex (^) is interpreted as a suspect marker and serves to mark the following character as suspect. For example, in Ne^vada, the v is marked as suspect. The value of these special characters is assessed when computing marked character efficiency.

Each tilde (~) in a correct text file is treated as a wildcard and allows zero or one arbitrary character to be generated for it without an error being charged. For example, suppose the page contains a character that a page reader is not expected to identify, such as a degree symbol (˚) or Greek letter (δ). Then the character should be represented in the correct file by a tilde so that a page reader may generate any single character for it, or no character at all, without penalty. For more information on the use of wildcards in ground truth, see reference [12].

Extraneous spacing characters are ignored in text files and have no effect on the computation of performance measures. Specifically, blank lines are disregarded, as well as leading and trailing blanks on a line. Multiple consecutive blanks within a line are treated as a single blank. The “newline” character at the end of each line is not ignored and appears as “<\n>” within the accuracy reports. Other whitespace characters, such as tabs and formfeeds, are treated as blanks. Page-reading systems and human ground-truth preparers may freely utilize spacing characters to format their text.

1.3 Zoning

We assume that an ordered set of zones has been defined for a page image, and that the coordinates of these zones are communicated to the page-reading system when it processes the image. The resulting OCR-generated text file contains the recognized text for zone 1, followed by the text for zone 2, and so on. The correct text file for this page must match this sequence, with the ground truth for zone 1 appearing first, followed by the ground truth for zone 2, and so on. Character and word accuracy are computed from these files as described in Sections 2 and 3. However, for the evaluation of automatic zoning, which is discussed in Section 4, the correspondence between OCR-generated and correct text is not guaranteed. In automatic zoning, the page reader processes the page image without the benefit of zone coordinates; hence, it may produce text in a sequence that does not match the correct file.

2 Character Accuracy

2.1 The **accuracy** Program

accuracy *correctfile* *generatedfile* [*accuracy_report*]

The **accuracy** program compares the correct text found in *correctfile* with the OCR-generated text found in *generatedfile*. A character accuracy report is written to *accuracy_report* if specified; otherwise, it is written to “stdout,” the Unix standard output file.

We will illustrate this program with an example. The following is a small page image containing two columns of text. Assume that zone 1 contains the left-hand column and zone 2 contains the right-hand column.

crushed under vacuum in stainless steel tubes. Liberated water was extracted at 200°C and converted. using uranium, into hydrogen for D/H analyses. The deuterium content is expressed in parts per thousand difference (per mil) relative to standard mean ocean water (SMOW) [normalized to the V-SMOW/SLAP scale (*7*)}. The δD values are plotted against age in Fig. 2.

We cannot attribute the changes in deu-terium to water-mineral exchange be-cause the water-bearing fractures in the regional carbonate aquifer, feeding the modern (and fossil) flow system, are typically coated with calcite or dolomite (*8*). This coating precludes the exchange of hydrogen between water and clay minerals during flow from recharge to discharge areas. In fact, the difference in

The following is the correct text for this page. Notice the two wildcards.

```
crushed under vacuum in stainless steel
tubes.  Liberated water was extracted at
200~C and converted, using uranium,
into hydrogen for D/H analyses.  The
deuterium content is expressed in parts
per thousand difference (per mil) relative
to standard mean ocean water (SMOW)
[normalized to the V-SMOW/SLAP
scale (7)].  The ~D values are plotted
against age in Fig. 2.
We cannot attribute the changes in deu-
terium to water-mineral exchange be-
cause the water-bearing fractures in the
regional carbonate aquifer, feeding the
modern (and fossil) flow system, are
typically coated with calcite or dolomite
(8).  This coating precludes the exchange
of hydrogen between water and clay
minerals during flow from recharge to
discharge areas.  In fact, the difference in
```

Here is OCR-generated text for this page. Notice the reject characters and suspect markers.

```
crushed under vacuum in stainless steel
tubes. Liberated water was extracted at
200”C and converted. using uranium,
into hydrogen for D/H analyses. The
deuterium content is expressed in parts
per thousand difference (per mil) relative
to standard mean ocean water (SMOW)
[normalized to the V-SMOWISLAP
scale (7)1. The 6D values are plotted
against age in Fig. 3.
We cannot attribute the changes in dcu-
terium to water-mineral exchange be-
cause the water-bearing ir^.acturss in the
regional carbonaie aquif^er, feeding the
modern (and fo~sil) now ~vstem. are
typically coated ~-^.ith calci~s or dolomite
(6). This coating pr-ecludcs the exchanjie
of hyd^l-ogen bet~^.etrn water and clay
minerals during now from ^I^.echarge to
discharge areas. in f,rct, the di~ference in
```

On the next page is the character accuracy report produced by the **accuracy** program when given these correct and generated files as inputs. A character accuracy report consists of six sections. The first section indicates the number of characters in the ground truth (756), the number of errors made by the page reader (39), and the character accuracy (94.84%). The second section gives the number of reject characters (6), suspect markers (7), and false marks (1), followed by information relating to marked character efficiency: if a user examines the marked characters (1.72% of the text) and corrects the marked errors, the character accuracy will increase (to 96.96%).

Errors are actually edit operations (character insertions, substitutions, and deletions) that are needed to correct the OCR-generated text. The third section of the character accuracy report gives a breakdown of marked errors, unmarked errors, and total errors by edit operation.

The fourth section shows the accuracy by character class. Here the ground-truth characters are divided into classes and the percentage of characters recognized in each class is reported. The total number of ground-truth characters missed (29) is always equal to the number of insertions plus substitutions.

The fifth section lists the “confusions” sorted by the number of errors charged for each. In this example, four errors were charged because n was generated for fl. Since this confusion requires only two edit operations to correct (one insertion and one substitution), then this confusion must have occurred twice to account for a total of four errors.

The sixth and last section of a character accuracy report provides a complete enumeration of the ground-truth characters.

```
UNLV-ISRI OCR Accuracy Report Version 5.1
-----------------------------------------
     756   Characters
      39   Errors
   94.84%  Accuracy

       6   Reject Characters
       7   Suspect Markers
       1   False Marks
    1.72%  Characters Marked
   96.96%  Accuracy After Correction

     Ins    Subst      Del   Errors
       0       10        6       16   Marked
       2       17        4       23   Unmarked
       2       27       10       39   Total

   Count   Missed   %Right
     117        0   100.00   ASCII Spacing Characters
      31        4    87.10   ASCII Special Symbols
       6        2    66.67   ASCII Digits
      24        1    95.83   ASCII Uppercase Letters
     578       22    96.19   ASCII Lowercase Letters
     756       29    96.16   Total

  Errors   Marked   Correct-Generated
       4        0   {fl}-{n}
       3        3   {w}-{~-.}
       2        2   {r}-{I.}
       2        2   {r}-{l-}
       2        2   {sy}-{~v}
       2        2   {te}-{~s}
       2        2   {w}-{~.}
       2        0   {,}-{.}
       2        0   {a}-{,r}
       2        0   {e}-{c}
       2        0   {e}-{tr}
       2        0   {g}-{ji}
       1        1   {f}-{~}
       1        1   {s}-{~}
       1        1   {}-{.}
       1        0   {/}-{I}
       1        0   {2}-{3}
       1        0   {8}-{6}
       1        0   {I}-{i}
       1        0   {]}-{1}
       1        0   {e}-{s}
       1        0   {f}-{i}
       1        0   {t}-{i}
       1        0   {}-{-}

   Count   Missed   %Right
      20        0   100.00   {<\n>}
      97        0   100.00   { }
       5        0   100.00   {(}
       5        0   100.00   {)}
       5        2    60.00   {,}
       5        0   100.00   {-}
       7        0   100.00   {.}
       2        1    50.00   {/}
       2        0   100.00   {0}
```

2	1	50.00	{2}
1	0	100.00	{7}
1	1	0.00	{8}
1	0	100.00	{A}
1	0	100.00	{C}
2	0	100.00	{D}
1	0	100.00	{F}
1	0	100.00	{H}
1	1	0.00	{I}
2	0	100.00	{L}
2	0	100.00	{M}
2	0	100.00	{O}
1	0	100.00	{P}
3	0	100.00	{S}
3	0	100.00	{T}
1	0	100.00	{V}
3	0	100.00	{W}
1	0	100.00	{[}
1	1	0.00	{]}
56	1	98.21	{a}
7	0	100.00	{b}
26	0	100.00	{c}
27	0	100.00	{d}
88	5	94.32	{e}
14	4	71.43	{f}
16	1	93.75	{g}
20	0	100.00	{h}
37	0	100.00	{i}
21	2	90.48	{l}
13	0	100.00	{m}
44	0	100.00	{n}
28	0	100.00	{o}
7	0	100.00	{p}
1	0	100.00	{q}
45	2	95.56	{r}
31	2	93.55	{s}
51	2	96.08	{t}
20	0	100.00	{u}
4	0	100.00	{v}
10	2	80.00	{w}
4	0	100.00	{x}
7	1	85.71	{y}
1	0	100.00	{z}

2.2 The **synctext** Program

synctext [–H] [–i] [–s] [–T] *textfile1* *textfile2* . . . > *resultfile*

The **synctext** program can be used to show the alignment of two or more text files. If exactly two input files are specified (one correct file and one OCR-generated file), then the alignment that is computed is the same as the one computed by the **accuracy** program. This allows the user to see where errors occurred within the text. The algorithm used to compute this alignment is described in reference [11].

If more than two input files are specified, or if the “–H” option is given, then the algorithm described in reference [13] is utilized to perform the alignment. This algorithm is also used by the **vote** program to align multiple text files (see Section 2.8).

The “–T” option selects yet another alignment algorithm which can find transposed matches between two input files (see reference [10]). This algorithm is used by the **editop** program for automatic zoning evaluation (see Section 4.1). The “–i” option specifies that the alignment is to be performed on a case insensitive basis (the default is case sensitive), and the “–s” option displays suspect markers in the output.

The following is the output of the **synctext** program when given the example correct and OCR-generated files as inputs. The characters upon which the input files agree are shown first, and everywhere there is a difference, a number in braces identifies a footnote below that indicates what the difference is.

```
==============================================================================

crushed under vacuum in stainless steel
tubes. Liberated water was extracted at
200{1}C and converted{2} using uranium,
into hydrogen for D/H analyses. The
deuterium content is expressed in parts
per thousand difference (per mil) relative
to standard mean ocean water (SMOW)
[normalized to the V-SMOW{3}SLAP
scale (7){4}. The {5}D values are plotted
against age in Fig. {6}.
We cannot attribute the changes in d{7}u-
terium to water-mineral exchange be-
cause the water-bearing {8}r{9}actur{10}s in the
regional carbona{11}e aquifer, feeding the
modern (and fo{12}sil) {13}ow {14}stem{15} are
typically coated {16}ith calci{17} or dolomite
({18}). This coating pr{19}eclud{20}s the exchan{21}e
of hyd{22}ogen bet{23}e{24}n water and clay
minerals during {25}ow from {26}echarge to
discharge areas. {27}n f{28}ct, the di{29}ference in

==============================================================================
{1}
Correct   {~}
Generated {“}
==============================================================================
{2}
Correct   {,}
Generated {.}
==============================================================================
{3}
Correct   {/}
Generated {I}
==============================================================================
{4}
Correct   {]}
Generated {1}
==============================================================================
{5}
Correct   {~}
Generated {6}
```

```
===============================================================================
{6}
Correct   {2}
Generated {3}
===============================================================================
{7}
Correct   {e}
Generated {c}
===============================================================================
{8}
Correct   {f}
Generated {i}
===============================================================================
{9}
Correct   {}
Generated {.}
===============================================================================
{10}
Correct   {e}
Generated {s}
===============================================================================
{11}
Correct   {t}
Generated {i}
===============================================================================
{12}
Correct   {s}
Generated {~}
===============================================================================
{13}
Correct   {fl}
Generated {n}
===============================================================================
{14}
Correct   {sy}
Generated {~v}
===============================================================================
{15}
Correct   {,}
Generated {.}
===============================================================================
{16}
Correct   {w}
Generated {~-.}
===============================================================================
{17}
Correct   {te}
Generated {~s}
===============================================================================
{18}
Correct   {8}
Generated {6}
===============================================================================
{19}
Correct   {}
Generated {-}
===============================================================================
{20}
Correct   {e}
Generated {c}
===============================================================================
```

```
{21}
Correct   {g}
Generated {ji}
==============================================================================
{22}
Correct   {r}
Generated {l-}
==============================================================================
{23}
Correct   {w}
Generated {~.}
==============================================================================
{24}
Correct   {e}
Generated {tr}
==============================================================================
{25}
Correct   {fl}
Generated {n}
==============================================================================
{26}
Correct   {r}
Generated {I.}
==============================================================================
{27}
Correct   {I}
Generated {i}
==============================================================================
{28}
Correct   {a}
Generated {,r}
==============================================================================
{29}
Correct   {f}
Generated {~}
==============================================================================
```

2.3 The **accsum** Program

accsum *accuracy_report1 accuracy_report2* . . . > *accuracy_report*

The **accsum** program combines two or more character accuracy reports to produce an aggregate report. While the results for a single page are interesting, tabulating the results for a set of pages yields important insights into the page-reading process.

On the next page is an aggregate character accuracy report that was produced by combining 175 individual reports using the **accsum** program. We see that there is a total of 293,493 characters on these 175 pages, and that the page reader made 5,916 errors for an overall character accuracy of 97.98%.

Given this large amount of data, some interesting observations can be made. For example, only 94.53% of the digits were recognized correctly versus 99.07% of the lowercase letters. Only 91.98% of the occurrences of the numeral `1` were correctly identified.

The list of confusions is very long so it has been truncated here. The most common confusion for this set of pages, contributing 236 errors, is the generation of a space where there should be none, which causes a word to be split (e.g., Nev ada). There are 43 errors attributed to the opposite case, i.e., no space is generated where there should be one, which causes two words to be joined (e.g., LasVegas).

The numeral zero (0) was generated for the letter O a total of 92 times on these pages; it is difficult also for humans to distinguish these symbols. The confusion charged 39 errors was a single sequence of 39 characters that was omitted by the page reader. An ellipsis (. . .) in a confusion indicates that a long sequence of characters has been truncated.

```
UNLV-ISRI OCR Accuracy Report Version 5.1
-----------------------------------------
  293493   Characters
    5916   Errors
   97.98%  Accuracy

     700   Reject Characters
       5   Suspect Markers
       5   False Marks
    0.24%  Characters Marked
   98.37%  Accuracy After Correction

     Ins    Subst      Del   Errors
      71      650      415     1136   Marked
     807     2699     1274     4780   Unmarked
     878     3349     1689     5916   Total

   Count   Missed   %Right
   47850      234    99.51   ASCII Spacing Characters
   14306      603    95.78   ASCII Special Symbols
   14341      784    94.53   ASCII Digits
   18737      768    95.90   ASCII Uppercase Letters
  198259     1838    99.07   ASCII Lowercase Letters
  293493     4227    98.56   Total

  Errors   Marked   Correct-Generated
     236        0   {}-{ }
      92        0   {O}-{0}
      90        0   {,}-{.}
      61        0   {}-{‘}
      52        0   {1}-{l}
      52        0   {}-{.}
      43        0   { }-{}
      39        0   {tion of<\n>temperature a...}-{}
      38        0   {.}-{,}
      37        0   {,}-{}
      31        0   {1}-{I}
      25        0   {-}-{ }
      24        0   {0}-{o}
      24        0   {2}-{z}
      24        0   {in}-{m}
      24        0   {l}-{1}
      21       21   {}-{~}
      21        0   {l}-{I}
```

```
      20        0   {96}-{%}
      20        0   {o}-{a}
...
   Count   Missed   %Right
    6064       23    99.62   {<\n>}
   41786      211    99.50   { }
     159        3    98.11   {“}
      13        6    53.85   {#}
       3        0   100.00   {$}
      56       13    76.79   {%}
       6        1    83.33   {&}
      70        3    95.71   {‘}
     883       46    94.79   {(}
     881       44    95.01   {)}
      75        6    92.00   {*}
      54       13    75.93   {+}
    3192      150    95.30   {,}
    2353      113    95.20   {-}
    5620       88    98.43   {.}
     224       17    92.41   {/}
    2513      110    95.62   {0}
    2806      225    91.98   {1}
    1797      122    93.21   {2}
    1328       41    96.91   {3}
    1109       57    94.86   {4}
    1131       54    95.23   {5}
    1021       65    93.63   {6}
     858       31    96.39   {7}
     812       35    95.69   {8}
     966       44    95.45   {9}
     234        4    98.29   {:}
     157        4    97.45   {;}
      25        0   100.00   {<}
     170       38    77.65   {=}
       6        2    66.67   {>}
      30        2    93.33   {?}
    1267       43    96.61   {A}
     512       22    95.70   {B}
    1393       31    97.77   {C}
     686       29    95.77   {D}
    1494       39    97.39   {E}
     668       14    97.90   {F}
     500       28    94.40   {G}
     542       19    96.49   {H}
    1097       38    96.54   {I}
     198        3    98.48   {J}
     177        7    96.05   {K}
     710       13    98.17   {L}
     774       55    92.89   {M}
    1135       45    96.04   {N}
     902      147    83.70   {O}
     811       38    95.31   {P}
      82        7    91.46   {Q}
    1187       41    96.55   {R}
    1546       41    97.35   {S}
    1656       31    98.13   {T}
     428       38    91.12   {U}
     279       11    96.06   {V}
     452       20    95.58   {W}
      67        2    97.01   {X}
     127        2    98.43   {Y}
```

```
   47       4   91.49   {Z}
   33      10   69.70   {[}
   33      12   63.64   {]}
    3       2   33.33   {`}
17028     111   99.35   {a}
 2638      50   98.10   {b}
 7269      75   98.97   {c}
 7500      32   99.57   {d}
25274     135   99.47   {e}
 4819      93   98.07   {f}
 3434      50   98.54   {g}
 7603      59   99.22   {h}
15387     167   98.91   {i}
  135      12   91.11   {j}
  851      16   98.12   {k}
 9304     156   98.32   {l}
 4966     100   97.99   {m}
14430     118   99.18   {n}
15151     107   99.29   {o}
 4371      40   99.08   {p}
  364       6   98.35   {q}
13574     107   99.21   {r}
12743      96   99.25   {s}
17702     103   99.42   {t}
 5627      58   98.97   {u}
 2023      22   98.91   {v}
 2216      39   98.24   {w}
  590      30   94.92   {x}
 2898      49   98.31   {y}
  362       7   98.07   {z}
    3       3    0.00   {{}
   20      20    0.00   {|}
    3       3    0.00   {}}
```

2.4 The **groupacc** Program

groupacc *groupfile accuracy_report* [*groupacc_report*]

In accuracy by character class, ground-truth characters are grouped according to predefined classes, such as the ASCII digits and the ASCII lowercase letters. The **groupacc** program allows user-defined groupings. This program summarizes the accuracy data from *accuracy_report* for the characters specified in *groupfile*. The results are written to *groupacc_report* if specified; otherwise, they are written to stdout.

For example, suppose we wish to focus on lowercase letters with descenders. We would create a group file containing the desired letters: gjpqy. Given this group file and the aggregate character accuracy report shown above, the **groupacc** program produces the following display:

```
Count   Missed   %Right
 3434       50    98.54   {g}
  135       12    91.11   {j}
 4371       40    99.08   {p}
```

```
   364        6    98.35   {q}
  2898       49    98.31   {y}
 11202      157    98.60   Total
```

2.5 The **accci** Program

accci *accuracy_report1 accuracy_report2* . . . > *resultfile*

Given a set of character accuracy reports as input, the **accci** program computes an approximate 95% confidence interval for character accuracy using the method of jackknife estimation (which is described in reference [14]). Each input report is treated as one observation. For best results, at least 30 observations are needed.

The following is the output from the **accci** program when given 175 character accuracy reports as input:

```
          175   Observations
       293493   Characters
         5916   Errors
        97.98%  Accuracy
 97.75%, 98.22%  Approximate 95% Confidence Interval for Accuracy
```

2.6 The **accdist** Program

accdist *accuracy_report1 accuracy_report2* . . . > *xyfile*

The **accdist** program computes the distribution of the character accuracies found in a set of character accuracy reports. The results are written in "*xy* format" to stdout. In this format, each line contains the *x*- and *y*-coordinates of a data point separated by spaces. A file in this format can be easily imported into most graphing programs.

The **accdist** program produces one data point for each value of *x* from 0 to 100. The *y* value is the percentage of characters recognized with at least *x*% character accuracy. Below is a portion of the output for the 175-page sample. It shows, for example, that 98.25% of the sample was recognized with at least 95% character accuracy. It is widely believed that it costs more to correct OCR-generated text that is less than 95% accurate than it costs to type the entire text from scratch. Thus, pages recognized with less than 95% accuracy should be sent to a manual data-entry operation. In this example, pages containing 1.75% of the characters should be entered manually.

```
  0 100.00
  1 100.00
...
 94  98.78
```

```
 95  98.25
 96  95.09
 97  88.51
 98  63.68
 99  16.99
100   0.53
```

2.7 The **ngram** Program

ngram [–n 1|2|3] *textfile1 textfile2* . . . > *resultfile*

The **ngram** program computes *n*-gram statistics for one or more text files. If the “–n” option is omitted, or if “–n 1” is specified, this program displays the frequency of each character found in the input files (“unigrams”). If “–n 2” is chosen, the frequency of every distinct character pair is shown (“bigrams”), and if “–n 3” is selected, the frequency of each unique triple of characters (i.e., three consecutive characters) is displayed (“trigrams”).

The *n*-gram statistics are displayed twice: first in the collating order of the characters, and then in order by decreasing frequency. Below is abbreviated unigram output for 175 correct text files. The blank character occurs most often, with 41,786 occurrences found in the input files. The letter `e` is the next most common, with 25,274 occurrences. In this example, none of the occurrences have been marked as suspect.

```
   Count  Suspect
    6064        0   {<\n>}
   41786        0   { }
     159        0   {“}
      13        0   {#}
       3        0   {$}
      56        0   {%}
       6        0   {&}
...
     590        0   {x}
    2898        0   {y}
     362        0   {z}
       3        0   {{}
      20        0   {|}
       3        0   {}}
    5453        0   {~}
  298946        0   Total

   Count  Suspect
   41786        0   { }
   25274        0   {e}
   17702        0   {t}
   17028        0   {a}
   15387        0   {i}
   15151        0   {o}
   14430        0   {n}
...
```

```
    13         0   {#}
     6         0   {&}
     6         0   {>}
     3         0   {$}
     3         0   {`}
     3         0   {{}
     3         0   {}}
298946         0   Total
```

2.8 The **vote** Program

vote [–O] [–o *outputfile*] [–s *m*/*n*] [–w *m*/*n*] *textfile1* *textfile2* . . .

Given OCR-generated text files that were produced by different page readers for the same page, the **vote** program applies a voting algorithm to produce a more accurate single text file. The input files are first aligned so that agreements and disagreements among the page readers are evident. Then a majority vote is taken to resolve the disagreements. For example, if three page readers believe that a character is an `e` and two believe it to be a `c`, then an `e` will be output by the voting algorithm. (The algorithm is actually more complicated than this, and involves heuristics for breaking ties.) The resulting text is written to *outputfile* if specified; otherwise, it is written to stdout.

The accuracy of the voting output is normally greater than the accuracy of the text from each page reader. ISRI tests have shown that the number of errors in voting output can be as much as 80% less than the number of errors made by the most accurate of the page readers. See references [1], [3], and [5] for voting test results.

The "–O" option enables some important optimizations and should be specified to get the best results. An output character is marked as suspect if it receives no more than the fraction of votes specified by the "–s" option. For example, output characters that receive one-third or less of the possible number of votes will be marked as suspect if "–s 1/3" is specified. The "–w" option indicates the fraction of a vote that each input character receives if it is marked as suspect. This reduces the influence of marked characters on the voting.

3 Word Accuracy

3.1 The **wordacc** Program

wordacc [–S *stopwordfile*] *correctfile* *generatedfile* [*wordacc_report*]

The **wordacc** program compares the correct text found in *correctfile* with the OCR-generated text found in *generatedfile*. A word accuracy report is written to *wordacc_report* if specified; otherwise, it is written to stdout.

Only the words found in *stopwordfile* are considered to be stopwords. (If the "–S" option is omitted, the default set of 110 stopwords from the BASISplus text retrieval system is used. See reference [15] for information on this system.) If stopwords are placed in the file in order by decreasing frequency of usage (as determined from some large corpus), then the file can also be used with the **nonstopacc** program (see Section 3.3). Here is an example of such a file containing 200 English stopwords in decreasing order of frequency:

```
the of and to a in that is was he for it with as his on be at by i this had not
are but from or have an they which one you were her all she there would their
we him been has when who will more no if out so said what up its about into
than them can only other new some could these two may then do first any my now
such like our over man me even most made after also did many before must
through back years where much your way well down should because each just those
mr how too state good very make still see men work long get here between both
being under never same another know while last might us great old year off come
since against go came right used take three states himself few use during
without again place around however small mrs thought went say part once general
high upon every does got number until always away something fact though less
put think almost enough far took yet better nothing end why find going asked
later knew point next give group toward young let room side given
```

On the next page is the word accuracy report produced by the **wordacc** program when given this stopword file and the example correct and generated files. A word accuracy report consists of seven sections. The first section indicates the number of words in the ground truth (119), the number of words misrecognized by the page reader (18), and the word accuracy (84.87%). The second and third sections show the stopword accuracy (92.86%) and the non-stopword accuracy (80.52%), respectively, with a breakdown by word length for each.

The fourth section presents the results for distinct non-stopword accuracy, which differs from non-stopword accuracy. See reference [4] for a description of this performance measure.

The fifth section gives the phrase accuracy for phrases of lengths 1 through 8. The sixth and seventh sections provide a complete enumeration of the stopwords and non-stopwords, respectively. Word accuracy is determined on a case-insensitive basis, so the words are displayed here in lowercase only.

```
UNLV-ISRI OCR Word Accuracy Report Version 5.1
----------------------------------------------
     119   Words
      18   Misrecognized
   84.87%  Accuracy

Stopwords
   Count   Missed   %Right   Length
      17        0   100.00        2
      16        0   100.00        3
       5        2    60.00        4
       1        0   100.00        5
       1        0   100.00        6
       2        1    50.00        7
      42        3    92.86    Total

Non-stopwords
   Count   Missed   %Right   Length
       5        0   100.00        1
       6        1    83.33        3
       7        4    42.86        4
      13        0   100.00        5
       8        2    75.00        6
      11        1    90.91        7
      12        3    75.00        8
      12        3    75.00        9
       3        1    66.67       10
      77       15    80.52    Total

Distinct Non-stopwords
   Count   Missed   %Right   Occurs
      58        9    84.48        1
       7        1    85.71        2
       1        0   100.00        5
      66       10    84.85    Total

Phrases
   Count   Missed   %Right   Length
     119       18    84.87        1
     118       31    73.73        2
     117       39    66.67        3
     116       47    59.48        4
     115       53    53.91        5
     114       57    50.00        6
     113       59    47.79        7
     112       61    45.54        8

Stopwords
   Count   Missed   %Right
       1        0   100.00   against
       3        0   100.00   and
       2        0   100.00   are
       1        0   100.00   at
       1        0   100.00   be
       1        1     0.00   between
       1        0   100.00   during
       1        1     0.00   fact
       1        0   100.00   for
       1        0   100.00   from
       7        0   100.00   in
```

1	0	100.00	into
1	0	100.00	is
1	0	100.00	of
1	0	100.00	or
9	0	100.00	the
1	0	100.00	this
4	0	100.00	to
1	0	100.00	under
1	0	100.00	was
1	0	100.00	we
1	1	0.00	with

Non-stopwords

Count	Missed	%Right	
1	0	100.00	age
1	0	100.00	analyses
1	0	100.00	aquifer
1	0	100.00	areas
1	0	100.00	attribute
1	0	100.00	bearing
1	0	100.00	c
1	1	0.00	calcite
1	0	100.00	cannot
1	1	0.00	carbonate
1	0	100.00	cause
1	0	100.00	changes
1	0	100.00	clay
1	0	100.00	coated
1	0	100.00	coating
1	0	100.00	content
1	0	100.00	converted
1	0	100.00	crushed
2	0	100.00	d
1	1	0.00	deu
1	0	100.00	deuterium
2	1	50.00	difference
1	0	100.00	discharge
1	0	100.00	dolomite
2	1	50.00	exchange
1	0	100.00	expressed
1	0	100.00	extracted
1	0	100.00	feeding
1	0	100.00	fig
2	2	0.00	flow
1	1	0.00	fossil
1	1	0.00	fractures
1	0	100.00	h
2	1	50.00	hydrogen
1	0	100.00	liberated
1	0	100.00	mean
1	0	100.00	mil
1	0	100.00	mineral
1	0	100.00	minerals
1	0	100.00	modern
1	0	100.00	normalized
1	0	100.00	ocean
1	0	100.00	parts
2	0	100.00	per
1	0	100.00	plotted
1	1	0.00	precludes
1	1	0.00	recharge

```
     1        0   100.00   regional
     1        0   100.00   relative
     1        0   100.00   scale
     1        1     0.00   slap
     2        1    50.00   smow
     1        0   100.00   stainless
     1        0   100.00   standard
     1        0   100.00   steel
     1        1     0.00   system
     1        0   100.00   terium
     1        0   100.00   thousand
     1        0   100.00   tubes
     1        0   100.00   typically
     1        0   100.00   uranium
     1        0   100.00   using
     1        0   100.00   v
     1        0   100.00   vacuum
     1        0   100.00   values
     5        0   100.00   water
```

3.2 The **wordaccsum** Program

wordaccsum *wordacc_report1* *wordacc_report2* . . . > *wordacc_report*

The **wordaccsum** program combines two or more word accuracy reports to produce an aggregate report. Below is the aggregate report that was produced by combining 175 individual reports. Of 43,928 words on these 175 pages, 2,211 were misrecognized for an overall word accuracy of 94.97%. It is usual to see a higher overall stopword accuracy (97.80%) than non-stopword accuracy (93.22%). The lists of stopwords and non-stopwords are very long and have been truncated here.

```
UNLV-ISRI OCR Word Accuracy Report Version 5.1
----------------------------------------------
   43928   Words
    2211   Misrecognized
   94.97%  Accuracy

Stopwords
   Count   Missed   %Right   Length
     810       46    94.32        1
    6069      128    97.89        2
    6065       92    98.48        3
    2438       61    97.50        4
     949       27    97.15        5
     204       10    95.10        6
     232        5    97.84        7
       1        0   100.00        9
   16768      369    97.80    Total

Non-stopwords
   Count   Missed   %Right   Length
    2275      250    89.01        1
    1050      249    76.29        2
```

```
    1425      169    88.14        3
    3025      160    94.71        4
    3705      186    94.98        5
    3327      172    94.83        6
    3216      139    95.68        7
    3132      166    94.70        8
    2345      125    94.67        9
    1560       96    93.85       10
    1032       57    94.48       11
     599       38    93.66       12
     260       17    93.46       13
     118        5    95.76       14
      58       11    81.03       15
      23        2    91.30       16
       3        0   100.00       17
       5        0   100.00       18
       2        0   100.00       19
   27160     1842    93.22    Total

Distinct Non-stopwords
   Count   Missed   %Right   Occurs
   11889      808    93.20        1
    2437       54    97.78        2
     868       15    98.27        3
     431        7    98.38        4
     241        1    99.59        5
     155        4    97.42        6
     107        4    96.26        7
      69        1    98.55        8
      62        1    98.39        9
      35        0   100.00       10
     112        1    99.11      >10
   16406      896    94.54    Total

Phrases
   Count   Missed   %Right   Length
   43928     2211    94.97        1
   43753     3927    91.02        2
   43578     5446    87.50        3
   43403     6812    84.31        4
   43228     8082    81.30        5
   43053     9249    78.52        6
   42878    10334    75.90        7
   42704    11351    73.42        8

Stopwords
   Count   Missed   %Right
     723       32    95.57   a
      69        1    98.55   about
      32        1    96.88   after
       9        1    88.89   again
      10        0   100.00   against
...

Non-stopwords
   Count   Missed   %Right
       2        2     0.00   ab
       3        0   100.00   abbreviations
       3        0   100.00   abernethy
       6        0   100.00   ability
       6        0   100.00   able
```

…

3.3 The **nonstopacc** Program

nonstopacc *stopwordfile* *wordacc_report* > *xyfile*

Given an ordered list of *N* stopwords in *stopwordfile* and a word accuracy report in *wordacc_report*, the **nonstopacc** program writes the data for the non-stopword accuracy curve to stdout. This curve presents non-stopword accuracy as a function of the number of stopwords. Examples of these curves can be found in reference [5]. The data points are written in *xy* format for values of *x* ranging from 0 to *N*. Each *y* value is the non-stopword accuracy when using the *x* most frequently-occurring stopwords from the stopword file. Here is a portion of the output for the example stopword file and aggregate word accuracy report:

```
  0  94.97
  1  94.69
  2  94.50
  3  94.35
  4  94.26
  5  94.23
  6  94.13
  7  94.09
  8  94.02
  9  94.00
...
200  93.22
```

3.4 The **wordaccci** Program

wordaccci *wordacc_report1* *wordacc_report2* . . . > *resultfile*

Given a set of word accuracy reports as input, the **wordaccci** program computes an approximate 95% confidence interval for word accuracy using the method of jackknife estimation [14]. This program is analogous to the **accci** program for character accuracy (see Section 2.5). The following is the output from the **wordaccci** program when given 175 word accuracy reports as input:

```
          175   Observations
        43928   Words
         2211   Misrecognized
        94.97%  Accuracy
94.40%, 95.54%  Approximate 95% Confidence Interval for Accuracy
```

3.5 The **wordaccdist** Program

wordaccdist *wordacc_report1 wordacc_report2* . . . > *xyfile*

The **wordaccdist** program computes the distribution of the word accuracies found in a set of word accuracy reports. The results are written in *xy* format to stdout. This program is analogous to the **accdist** program for character accuracy (see Section 2.6).

The **wordaccdist** program produces one data point for each value of *x* from 0 to 100. The *y* value is the percentage of words recognized with at least *x*% word accuracy.

3.6 The **wordfreq** Program

wordfreq *textfile1 textfile2* . . . > *resultfile*

The **wordfreq** program determines the frequency of words in one or more text files. The frequency data are displayed twice: first in the collating order of the words, and then in order by decreasing frequency. Below is abbreviated **wordfreq** output for 175 correct text files. The word “the” occurs most often, with 2,981 occurrences found in the input files.

```
   Count
     723   a
       2   ab
       3   abbreviations
       3   abernethy
       6   ability
...
      28   zone
       9   zones
       2   zonplt
       1   zro
       3   zubovic
   43928   Total

   Count
    2981   the
    1907   of
    1366   and
    1048   in
     723   a
...
       1   zirconates
       1   zirconium
       1   zirconolite
       1   zonal
       1   zro
   43928   Total
```

4 Automatic Zoning

4.1 The **editop** Program

editop *correctfile* *generatedfile* [*editop_report*]

The **editop** program compares the correct text found in *correctfile* with the OCR-generated text found in *generatedfile*. An edit operation report is written to *editop_report* if specified; otherwise, it is written to stdout. This program is used to evaluate the automatic zoning capability of a page reader.

In automatic zoning, the page reader attempts to locate all text regions on the page and determine their correct reading order. Character recognition is then performed for each text region that has been identified. The OCR-generated text reflects a combination of automatic zoning errors and character recognition errors.

The **editop** program estimates the number of edit operations needed to correct the OCR-generated text. Three types of edit operations are considered: character insertions, character deletions, and block move operations. If a page reader fails to find a text region, insertions are needed to enter the missing text; if it misidentifies a graphic region as text, it may generate extraneous characters that need to be deleted; and if it incorrectly determines the reading order, block move operations are required to re-order the text.

If a page reader performs automatic zoning on the example page and fails to identify the two columns, it may produce output such as this:

```
crushed under vacuum in stainless steel We cannot attribute the changes in dcu-
tubes. Liberated water was extracted at terium to water-mineral exchange be-
200”C and converted. using uranium, cause the water-bearing ir^.acturss in the
into hydrogen for D/H analyses. The regional carbonaie aquif^er, feeding the
deuterium content is expressed in parts modern (and fo~sil) now ~vstem. are
per thousand difference (per mil) relative typically coated ~-^.ith calci~s or dolomite
to standard mean ocean water (SMOW) (8). This coating pr-ecludcs the exchanjie
[normalized to the V-SMOWISLAP of hyd^l-ogen bet~^.etrn water and clay
scale (7)1. The 6D values are plotted minerals during now from ^I^.echarge to
against age in Fig. 3. discharge areas. in f,rct, the di~ference in
```

On the next page is the edit operation report produced by the **editop** program when given the correct file and this generated file as inputs. An edit operation report consists of two sections. The first section gives the number of insertions (30), deletions (40), and move operations (21) for correcting the generated text. The second section gives a breakdown of the move operations by length. In this example, there were 11 single-character moves, one move of 22 characters, one move of 30 characters, etc. (The count shown for length 100 is actually the number of moves of length 100 or more.)

```
UNLV-ISRI Edit Operation Report Version 5.1
-------------------------------------------
      30   Insertions
      40   Deletions
      21   Moves

Moves
   Count   Length
      11        1
       1       22
       1       30
       1       33
       1       34
       1       37
       1       38
       1       39
       1       69
       1       76
       1      100
```

4.2 The **editopsum** Program

editopsum *editop_report1 editop_report2* . . . > *editop_report*

The **editopsum** program combines two or more edit operation reports to produce an aggregate report. Below is the aggregate report that was produced by combining 175 individual reports. The distribution of the lengths of the 2,206 moves has been abbreviated here. There are 68 moves of length 100 or more.

```
UNLV-ISRI Edit Operation Report Version 5.1
-------------------------------------------
    2926   Insertions
   28787   Deletions
    2206   Moves

Moves
   Count   Length
    1543        1
     204        2
      97        3
      49        4
      28        5
      25        6
       9        7
       7        8
       7        9
       6       10
...
       1       77
       1       79
       1       84
       2       96
      68      100
```

4.3 The **editopcost** Program

editopcost *editop_report* [*editop_report2*] > *xyfile*

The **editopcost** program computes the cost of the edit operations described in *editop_report*, less the cost of the edit operations described in *editop_report2*, if specified. The cost 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. See reference [10] for details. The output is written to stdout in *xy* format. There is a data point specifying a threshold value (*x*) and the associated cost (*y*) for each threshold in the range 0 to 100.

Normally, *editop_report* 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 the automatic zoning errors and the character recognition errors. Hence, it is often desirable to specify a second aggregate report, *editop_report2*, that indicates the edit operations for correcting the OCR-generated text that was produced using manually-defined zones for the same set of pages. The cost of these operations is the cost of correcting only the character recognition errors. Deducting this cost from the total yields the cost of correcting the automatic zoning errors.

Below is an abbreviated example of **editopcost** output. When plotted, these data points form a curve showing the cost of correcting automatic zoning errors as a function of the threshold value. Examples of such curves can be seen in references [3], [4], and [5]. The costs given in the **editopcost** output are unnormalized but can be normalized by dividing each *y* value by the total number of ground-truth characters.

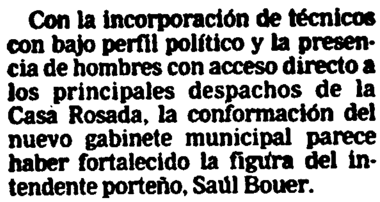
```
  0        77
  1      1010
  2      1459
  3      1803
  4      2088
...
 96     11710
 97     11767
 98     11824
 99     11881
100     11938
```

5 Foreign-Language OCR Testing

5.1 Latin1 Testing

The examples in the preceding sections illustrate how to use the ISRI Analytic Tools for evaluating English OCR using ASCII text files. These tools can also be used with Latin1 text files to evaluate OCR systems for any of the following languages: Danish, Dutch, Faroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish. Latin1 is the ISO 8859-1 standard 8-bit character encoding [16]. This encoding contains the 7-bit ASCII standard as a subset.

In this section, we illustrate how the Analytic Tools can be used to evaluate Spanish OCR. Here is a small page image containing Spanish text:


Con la incorporación de técnicos con bajo perfil político y la presencia de hombres con acceso directo a los principales despachos de la Casa Rosada, la conformación del nuevo gabinete municipal parece haber fortalecido la figura del intendente porteño, Saúl Bouer.

Latin1 is needed to represent the correct text because of the accented characters:

```
Con la incorporación de técnicos
con bajo perfil político y la presen-
cia de hombres con acceso directo a
los principales despachos de la
Casa Rosada, la conformación del
nuevo gabinete municipal parece
haber fortalecido la figura del in-
tendente porteño, Saúl Bouer.
```

Here is OCR-generated text for this page, also in Latin1:

```
Con la incorporación de técnicos
con bajo perfil político y la pre^sen~
cia de hombres con acceso directo a
los principales despachos de la
Cas^a Rosada, la conformación del
nuevo gabinete municipal parece
haber fortalecido la figura del i^ii^.
tendente porteño, Sañí Boner.
```

On the next page is the character accuracy report produced by the **accuracy** program when given these correct and generated files as inputs.

```
UNLV-ISRI OCR Accuracy Report Version 5.1
-----------------------------------------
     270   Characters
       7   Errors
   97.41%  Accuracy

       1   Reject Characters
       4   Suspect Markers
       2   False Marks
    1.85%  Characters Marked
   98.89%  Accuracy After Correction

     Ins    Subst      Del   Errors
       0        3        1        4   Marked
       0        3        0        3   Unmarked
       0        6        1        7   Total

   Count   Missed   %Right
      43        0   100.00   ASCII Spacing Characters
       5        2    60.00   ASCII Special Symbols
       5        0   100.00   ASCII Uppercase Letters
     211        3    98.58   ASCII Lowercase Letters
       6        1    83.33   Latin1 Lowercase Letters
     270        6    97.78   Total

  Errors   Marked   Correct-Generated
       3        3   {n-}-{ii.}
       2        0   {úl}-{ñí}
       1        1   {-}-{~}
       1        0   {u}-{n}

   Count   Missed   %Right
       8        0   100.00   {<\n>}
      35        0   100.00   { }
       2        0   100.00   {,}
       2        2     0.00   {-}
       1        0   100.00   {.}
       1        0   100.00   {B}
       2        0   100.00   {C}
       1        0   100.00   {R}
       1        0   100.00   {S}
      24        0   100.00   {a}
       4        0   100.00   {b}
      18        0   100.00   {c}
      10        0   100.00   {d}
      25        0   100.00   {e}
       4        0   100.00   {f}
       2        0   100.00   {g}
       3        0   100.00   {h}
      16        0   100.00   {i}
       1        0   100.00   {j}
      14        1    92.86   {l}
       3        0   100.00   {m}
      16        1    93.75   {n}
      23        0   100.00   {o}
      10        0   100.00   {p}
      14        0   100.00   {r}
      10        0   100.00   {s}
       8        0   100.00   {t}
       4        1    75.00   {u}
       1        0   100.00   {v}
```

```
       1        0   100.00   {y}
       1        0   100.00   {é}
       1        0   100.00   {í}
       1        0   100.00   {ñ}
       2        0   100.00   {ó}
       1        1     0.00   {ú}
```

The following is the output from the **synctext** program when given these correct and generated files as inputs:

```
================================================================================

Con la incorporación de técnicos
con bajo perfil político y la presen{1}
cia de hombres con acceso directo a
los principales despachos de la
Casa Rosada, la conformación del
nuevo gabinete municipal parece
haber fortalecido la figura del i{2}
tendente porteño, Sa{3} Bo{4}er.

================================================================================
{1}
Correct   {-}
Generated {~}
================================================================================
{2}
Correct   {n-}
Generated {ii.}
================================================================================
{3}
Correct   {úl}
Generated {ñí}
================================================================================
{4}
Correct   {u}
Generated {n}
================================================================================
```

The **wordacc** program needs a stopword file. Here is a file containing 200 Spanish stopwords in decreasing order of frequency (courtesy of Chris Buckley at Cornell):

```
de la el y en que a los del se por un con las para al una su no es lo como más
sus pero dijo este ya fue esta entre ha también dos son o está sin le sobre si
ser cuando hasta porque tiene donde desde parte sólo han todo muy hoy durante
hay tres quien están uno así todos después además otros expresó hace nuevo
ahora agregó primera hacer ante señaló les ese e será puede vez ayer mismo
tienen fueron cada contra aunque pasado mayor lugar otro antes nos mientras esa
esto ellos algunos primer gran tanto sido otra indicó nuevos eso bien menos
estos cuatro explicó embargo tener ni debe otras mejor había momento cual
informó era mucho luego hecho sino nueva pues sea quienes dentro qué cuenta
cinco me va según unos manera él comentó dar nada muchos sí aún pueden estar
siempre poco todas haber aquí tan segundo hizo ver toda fin yo casi podría
estas hacia seis algunas próximo aseguró decir bajo fuera varios misma
cualquier total estamos algo nosotros añadió mi grandes estaba ello través dio
ex afirmó tal tenemos existe últimos conocer respecto sería van dice primero
segunda cosas actualmente
```

Below is the word accuracy report produced by the **wordacc** program when given this stopword file, the correct file, and the OCR-generated file:

```
UNLV-ISRI OCR Word Accuracy Report Version 5.1
----------------------------------------------
      43   Words
       3   Misrecognized
   93.02%  Accuracy

Stopwords
   Count   Missed   %Right   Length
       2        0   100.00        1
       8        0   100.00        2
       6        0   100.00        3
       1        0   100.00        4
       2        0   100.00        5
      19        0   100.00    Total

Non-stopwords
   Count   Missed   %Right   Length
       1        1     0.00        2
       1        0   100.00        3
       2        1    50.00        4
       1        1     0.00        5
       6        0   100.00        6
       3        0   100.00        7
       4        0   100.00        8
       2        0   100.00        9
       2        0   100.00       11
       1        0   100.00       12
       1        0   100.00       13
      24        3    87.50    Total

Distinct Non-stopwords
   Count   Missed   %Right   Occurs
      24        3    87.50        1
      24        3    87.50    Total

Phrases
   Count   Missed   %Right   Length
      43        3    93.02        1
      42        4    90.48        2
      41        5    87.80        3
      40        5    87.50        4
      39        5    87.18        5
      38        5    86.84        6
      37        5    86.49        7
      36        5    86.11        8
```

Stopwords

Count	Missed	%Right	
1	0	100.00	a
1	0	100.00	bajo
3	0	100.00	con
3	0	100.00	de
2	0	100.00	del
1	0	100.00	haber
5	0	100.00	la
1	0	100.00	los
1	0	100.00	nuevo
1	0	100.00	y

Non-stopwords

Count	Missed	%Right	
1	0	100.00	acceso
1	1	0.00	bouer
1	0	100.00	casa
1	0	100.00	cia
1	0	100.00	conformación
1	0	100.00	despachos
1	0	100.00	directo
1	0	100.00	figura
1	0	100.00	fortalecido
1	0	100.00	gabinete
1	0	100.00	hombres
1	1	0.00	in
1	0	100.00	incorporación
1	0	100.00	municipal
1	0	100.00	parece
1	0	100.00	perfil
1	0	100.00	político
1	0	100.00	porteño
1	0	100.00	presen
1	0	100.00	principales
1	0	100.00	rosada
1	1	0.00	saúl
1	0	100.00	tendente
1	0	100.00	técnicos

5.2 Unicode Testing

The Unicode standard specifies a 16-bit character encoding for nearly all of the world's languages [17]. It includes Latin1 and ASCII as subsets. By using the Unicode representation of characters, the ISRI Analytic Tools can be utilized to evaluate OCR systems for almost any language in the world.

Each program of the Analytic Tools operates on files in "Extended ASCII" format. In this format, ASCII and Latin1 characters are stored directly in the file, and each 16-bit Unicode symbol is represented by four hexadecimal digits surrounded by angle brackets. For example, the sequence **AÄΩא** is stored as AÄ<03A9><05D0> because **A** is an ASCII character, **Ä** is a Latin1 character, and **Ω** and **א** are represented by codes 03A9 and 05D0 respectively in Unicode.

While all operations are performed using Extended ASCII files, the **asc2uni** and **uni2asc** programs are available to convert to and from the standard Unicode file format.

asc2uni < *ASCII_file* > *Unicode_file*
uni2asc < *Unicode_file* > *ASCII_file*

For example, the **accuracy** program accepts a correct file and an OCR-generated file in Extended ASCII format and produces a character accuracy report, also in Extended ASCII format. If the correct text was entered using a Unicode editor into a standard Unicode file, then an equivalent Extended ASCII version of this text can be created using the **uni2asc** program. Similarly, if an OCR system produces a standard Unicode file as output, then this generated text can be converted to Extended ASCII using the **uni2asc** program. The **asc2uni** program allows the character accuracy report to be converted to the standard Unicode file format for viewing with a Unicode display tool.

All of the programs presented in Sections 2 and 4 can be used with Extended ASCII files and work in the same way as described in those sections. However, the programs in Section 3 define a word to be any sequence of one or more ASCII or Latin1 letters. Hence, for languages that compose words using other symbols, word accuracy is unavailable.

We illustrate Unicode testing with an example in Japanese. At the top of the next page is a small page image containing Japanese text.

イメージスキャナで読み取った日
本語活字文書を、そのまま文字デ
ータに高速変換。今までのパソコ
ンOCRでは見られない高機能
を、ソフトウェアと拡張スロット用
ボードで実現。お手持ちのパソコ
ンで、膨大なデータの入力を迅速
に処理でき、文書入力のためのコ
ストと負担を大幅に軽減します。

Here is the correct text for this page in Extended ASCII:

<30A4><30E1><30FC><30B8><30B9><30AD><30E3><30CA><3067><8AAD><307F><53D6><3063><305F><65E5>
<672C><8A9E><6D3B><5B57><6587><66F8><3092><3001><305D><306E><307E><307E><6587><5B57><30C7>
<30FC><30BF><306B><9AD8><901F><5909><63DB><3002><4ECA><307E><3067><306E><30D1><30BD><30B3>
<30F3>OCR<3067><306F><898B><3089><308C><306A><3044><9AD8><6A5F><80FD>
<3092><3001><30BD><30D5><30C8><30A6><30A7><30A2><3068><62E1><5F35><30B9><30ED><30C3><30C8><7528>
<30DC><30FC><30C9><3067><5B9F><73FE><3002><304A><624B><6301><3061><306E><30D1><30BD><30B3>
<30F3><3067><3001><81A8><5927><306A><30C7><30FC><30BF><306E><5165><529B><3092><8FC5><901F>
<306B><51E6><7406><3067><304D><3001><6587><66F8><5165><529B><306E><305F><3081><306E><30B3>
<30B9><30C8><3068><8CA0><62C5><3092><5927><5E45><306B><8EFD><6E1B><3057><307E><3059><3002>

Here is a Unicode display of this correct text:

イメージスキャナで読み取った日
本語活字文書を、そのまま文字デ
ータに高速変換。今までのパソコ
ンOCRでは見られない高機能
を、ソフトウェアと拡張スロット用
ボードで実現。お手持ちのパソコ
ンで、膨大なデータの入力を迅速
に処理でき、文書入力のためのコ
ストと負担を大幅に軽減します。

Here is a Unicode display of OCR-generated text for this page:

イメージスキャナで読み取った日
本語活字文書在、そのまま文字テ
ータ|二高連変換。 含までの／（、ノコ
ン・ｃ円では見られなし｜高機能
在、 ソワトウウフと拡張スロット用
ボートで実現。 お手持ちの／く、ノコ
ンで、一大なデ・タの人力在Ｂ速
|二処理でと文善人力のためのコ
ストと員担在大福に軽高します．

The next three pages show a Unicode display of the character accuracy report produced by the **accuracy** program for this correct and generated text.

```
UNLV-ISRI OCR Accuracy Report Version 5.1
-----------------------------------------
     144   Characters
      52   Errors
  63.89%   Accuracy

       0   Reject Characters
       0   Suspect Markers
       0   False Marks
   0.00%   Characters Marked
  63.89%   Accuracy After Correction

     Ins    Subst      Del   Errors
       0        0        0        0   Marked
       1       41       10       52   Unmarked
       1       41       10       52   Total

   Count   Missed   %Right
       9        0   100.00   ASCII Spacing Characters
       3        3     0.00   ASCII Uppercase Letters
       7        1    85.71   CJK Symbols and Punctuation
      42        8    80.95   Hiragana
      38       22    42.11   Katakana
      45        8    82.22   CJK Unified Ideographs
     144       42    70.83   Total

  Errors   Marked   Correct-Generated
       4        0   {パソ}-{／く、ノ}
       4        0   {パソ}-{／（、ノ}
       4        0   {ータに}-{ー夕|二}
       3        0   {を}-{在}
       3        0   {ロット}-{ロツト}
       3        0   {OCR}-{°ｃ円}
       2        0   {い}-{Ｌl}
       2        0   {き、}-{と}
       2        0   {に}-{|二}
       2        0   {を迅}-{在Ｂ}
       2        0   {ェア}-{ウフ}
       2        0   {フト}-{ワト}
       2        0   {ージ}-{ーシ}
       2        0   {ータ}-{・夕}
       2        0   {ード}-{ート}
       2        0   {今}-{ 含}
       2        0   {}-{ }
       1        0   {デ}-{テ}
       1        0   {ト}-{ト}
       1        0   {ャ}-{ヤ}
       1        0   {幅}-{福}
       1        0   {書}-{善}
       1        0   {減}-{高}
       1        0   {膨}-{一}
       1        0   {負}-{員}
       1        0   {速}-{連}
```

Count	Missed	%Right	
9	0	100.00	{<\n>}
1	1	0.00	{C}
1	1	0.00	{O}
1	1	0.00	{R}
4	1	75.00	{、}
3	0	100.00	{。}
1	1	0.00	{い}
1	0	100.00	{お}
1	1	0.00	{き}
1	0	100.00	{し}
1	0	100.00	{す}
1	0	100.00	{そ}
2	0	100.00	{た}
1	0	100.00	{ち}
1	0	100.00	{っ}
6	0	100.00	{で}
2	0	100.00	{と}
2	0	100.00	{な}
3	2	33.33	{に}
6	0	100.00	{の}
1	0	100.00	{は}
4	0	100.00	{ま}
1	0	100.00	{み}
1	0	100.00	{め}
1	0	100.00	{ら}
1	0	100.00	{れ}
4	4	0.00	{を}
1	1	0.00	{ア}
1	0	100.00	{イ}
1	0	100.00	{ウ}
1	1	0.00	{ェ}
1	0	100.00	{キ}
3	0	100.00	{コ}
1	1	0.00	{ジ}
3	0	100.00	{ス}
3	2	33.33	{ソ}
2	2	0.00	{タ}
1	1	0.00	{ッ}
2	1	50.00	{デ}
3	3	0.00	{ト}
1	1	0.00	{ド}
1	0	100.00	{ナ}
2	2	0.00	{パ}

1	1	0.00	{フ}
1	0	100.00	{ボ}
1	0	100.00	{メ}
1	1	0.00	{ャ}
1	1	0.00	{ロ}
2	0	100.00	{ン}
4	4	0.00	{一}
1	1	0.00	{今}
2	0	100.00	{入}
1	0	100.00	{処}
2	0	100.00	{力}
1	0	100.00	{取}
1	0	100.00	{変}
2	0	100.00	{大}
2	0	100.00	{字}
1	0	100.00	{実}
1	1	0.00	{幅}
1	0	100.00	{張}
1	0	100.00	{手}
1	0	100.00	{担}
1	0	100.00	{拡}
1	0	100.00	{持}
1	0	100.00	{換}
3	0	100.00	{文}
1	0	100.00	{日}
2	1	50.00	{書}
1	0	100.00	{本}
1	0	100.00	{機}
1	0	100.00	{活}
1	1	0.00	{減}
1	0	100.00	{現}
1	0	100.00	{理}
1	0	100.00	{用}
1	0	100.00	{能}
1	1	0.00	{膨}
1	0	100.00	{見}
1	0	100.00	{語}
1	0	100.00	{読}
1	1	0.00	{負}
1	0	100.00	{軽}
1	1	0.00	{迅}
2	1	50.00	{速}
2	0	100.00	{高}

Here is the output from the **synctext** program for this correct and generated text:

```
========================================================================

イメ{1}スキ{2}ナで読み取った日
本語活字文書{3}、そのまま文字{4}
{5}高{6}変換。{7}までの{8}コ
ン{9}では見られな{10}高機能
{11}、{12}ソ{13}ウ{14}と拡張ス{15}用
ポ{16}で実現。{17}お手持ちの{18}コ
ンで、{19}大なデ{20}の人力{21}速
{22}処理で{23}文{24}人力のためのコ
ス{25}と{26}担{27}大{28}に軽{29}します.

========================================================================
{1}
Correct   {ージ}
Generated {ーシ}
========================================================================
{2}
Correct   {ャ}
Generated {ヤ}
========================================================================
{3}
Correct   {を}
Generated {在}
========================================================================
{4}
Correct   {デ}
Generated {テ}
========================================================================
{5}
Correct   {ータに}
Generated {ータ|二}
========================================================================
{6}
Correct   {速}
Generated {連}
========================================================================
{7}
Correct   {今}
Generated { 含}
========================================================================
```

```
{8}
Correct   {パソ}
Generated {／（、ノ}
=====================================================================
{9}
Correct   {OCR}
Generated {° c 円}
=====================================================================
{10}
Correct   {い}
Generated {L l}
=====================================================================
{11}
Correct   {を}
Generated {在}
=====================================================================
{12}
Correct   {}
Generated { }
=====================================================================
{13}
Correct   {フト}
Generated {ワト}
=====================================================================
{14}
Correct   {ェア}
Generated {ウフ}
=====================================================================
{15}
Correct   {ロット}
Generated {ロツト}
=====================================================================
{16}
Correct   {ード}
Generated {ート}
=====================================================================
{17}
Correct   {}
Generated { }
=====================================================================
{18}
Correct   {パソ}
Generated {／く、ノ}
=====================================================================
```

```
{19}
Correct   {膨}
Generated {一}
=====================================================================
{20}
Correct   {一タ}
Generated {・タ}
=====================================================================
{21}
Correct   {を迅}
Generated {在B}
=====================================================================
{22}
Correct   {に}
Generated {|二}
=====================================================================
{23}
Correct   {き、}
Generated {と}
=====================================================================
{24}
Correct   {書}
Generated {善}
=====================================================================
{25}
Correct   {ト}
Generated {ト}
=====================================================================
{26}
Correct   {負}
Generated {員}
=====================================================================
{27}
Correct   {を}
Generated {在}
=====================================================================
{28}
Correct   {幅}
Generated {福}
=====================================================================
{29}
Correct   {減}
Generated {高}
=====================================================================
```

6 Source Files

Version 5.1 of the ISRI Analytic Tools consists of the 19 programs described in this report. There is one C source file for each program:

accci.c	editop.c	nonstopacc.c	wordaccci.c
accdist.c	editopcost.c	synctext.c	wordaccdist.c
accsum.c	editopsum.c	uni2asc.c	wordaccsum.c
accuracy.c	groupacc.c	vote.c	wordfreq.c
asc2uni.c	ngram.c	wordacc.c	

Each program is linked with a library of 15 modules. There is a ".c" file and a ".h" include file for each module. For example, the accrpt module consists of the files accrpt.c and accrpt.h. Here is a brief description of these modules:

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

References

[1] S. V. Rice, J. Kanai, and T. A. Nartker. A report on the accuracy of OCR devices. Technical Report 92-02, Information Science Research Institute, University of Nevada, Las Vegas, March 1992.

[2] S. V. Rice, J. Kanai, and T. A. Nartker. An evaluation of OCR accuracy. Technical Report 93-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1993.

[3] S. V. Rice, J. Kanai, and T. A. Nartker. The third annual test of OCR accuracy. Technical Report 94-03, Information Science Research Institute, University of Nevada, Las Vegas, April 1994.

[4] S. V. Rice, F. R. Jenkins, and T. A. Nartker. The fourth annual test of OCR accuracy. Technical Report 95-04, Information Science Research Institute, University of Nevada, Las Vegas, April 1995.

[5] S. V. Rice, F. R. Jenkins, and T. A. Nartker. The fifth annual test of OCR accuracy. Technical Report 96-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1996.

[6] T. A. Nartker, S. V. Rice, and J. Kanai. OCR accuracy: UNLV's second annual test. *Inform*, Association for Information and Image Management, 8(1):40+, January 1994.

[7] T. A. Nartker and S. V. Rice. OCR accuracy: UNLV's third annual test. *Inform*, Association for Information and Image Management, 8(8):30+, September 1994.

[8] T. A. Nartker, S. V. Rice, and F. R. Jenkins. OCR accuracy: UNLV's fourth annual test. *Inform*, Association for Information and Image Management, 9(7):38+, July 1995.

[9] T. A. Nartker, S. V. Rice, and F. R. Jenkins. OCR accuracy: UNLV's fifth annual test. *Inform*, Association for Information and Image Management, to appear, 1996.

[10] 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.

[11] S. V. Rice. *Measuring the Accuracy of Page-Reading Systems*. Ph.D. dissertation,

University of Nevada, Las Vegas, 1996.

[12] S. V. Rice, J. Kanai, and T. A. Nartker. Preparing OCR test data. Technical Report 93-08, Information Science Research Institute, University of Nevada, Las Vegas, June 1993.

[13] 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.

[14] E. J. Dudewicz and S. N. Mishra. *Modern Mathematical Statistics*, pages 743-748. John Wiley & Sons, 1988.

[15] Information Dimensions, Inc., Dublin, Ohio. *BASISplus Database Administration Reference, Release L*, June 1990.

[16] International Organization for Standardization, Geneva, Switzerland. *Information Processing — 8-bit Single-byte Coded Graphic Character Sets: Part 1. Latin Alphabet No. 1*, 1987.

[17] The Unicode Consortium, San Jose, California. *The Unicode Standard: Worldwide Character Encoding Version 1.0*, Addison-Wesley, Volume 1, 1991, and Volume 2, 1992.