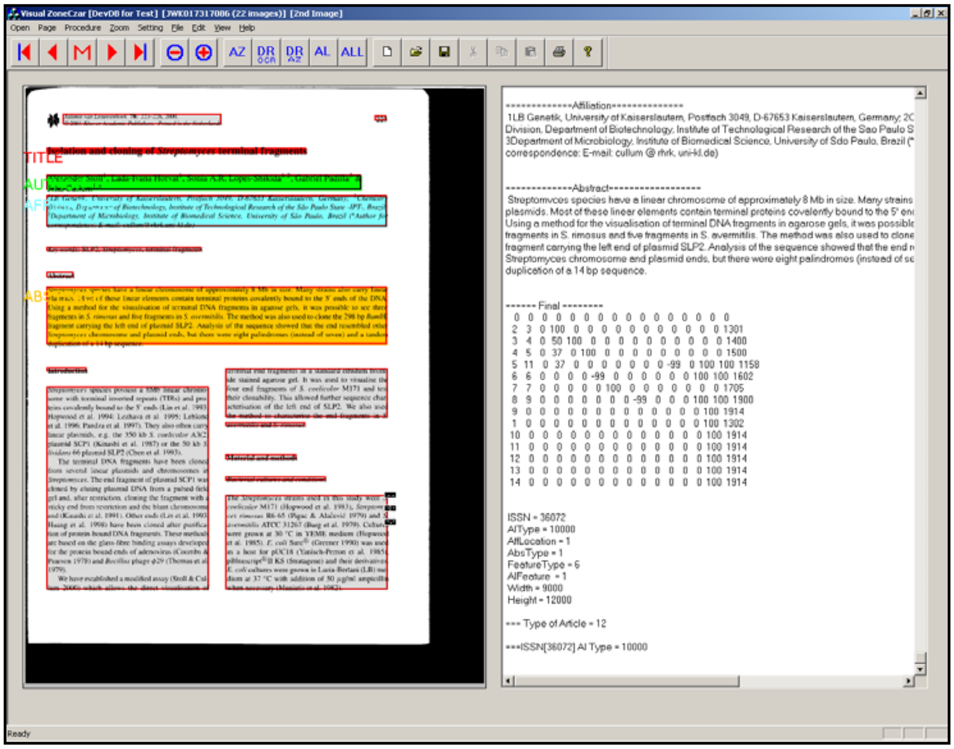
Automating the production of bibliographic records for MEDLINE


Visual ZoneCzar [DevDB for Test] [JWK017317086 (22 images)] [2nd Image]
Open Page Procedure Zoom Setting File Edit View Help
M
AZ
DR OCR
DR AZ
AL
ALL
TITLE
AU
AF
AB
Ready

```
==============Affiliation==============
 1LB Genetik, University of Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern, Germany; 2C
Division, Department of Biotechnology, Institute of Technological Research of the Sao Paulo S
3Department of Microbiology, Institute of Biomedical Science, University of Sdo Paulo, Brazil (*
correspondence: E-mail: cullum @ rhrk. uni-kl.de)


==============Abstract==================
 Streptomvces species have a linear chromosome of approximately 8 Mb in size. Many strains
plasmids. Most of these linear elements contain terminal proteins covalently bound to the 5' en
Using a method for the visualisation of terminal DNA fragments in agarose gels, it was possibl
fragments in S. rimosus and five fragments in S. avermitilis. The method was also used to clone
fragment carrying the left end of plasmid SLP2. Analysis of the sequence showed that the end r
Streptomyces chromosome and plasmid ends, but there were eight palindromes (instead of se
duplication of a 14 bp sequence.


------ Final --------
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  2   3   0 100   0   0   0   0   0   0   0   0   0   0   0 1301
  3   4   0  50 100   0   0   0   0   0   0   0   0   0   0 1400
  4   5   0  37   0 100   0   0   0   0   0   0   0   0   0 1500
  5  11   0  37   0   0   0   0   0   0   0 -99   0 100 100 1158
  6   6   0   0   0   0 -99   0   0   0   0   0   0 100 100 1602
  7   7   0   0   0   0   0 100   0   0   0   0   0   0   0 1705
  8   9   0   0   0   0   0   0   0 -99   0   0   0 100 100 1900
  9   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
  1   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1302
 10   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 11   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 12   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 13   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 14   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914


 ISSN = 36072
 AlType = 10000
 AffLocation = 1
 AbsType = 1
 FeatureType = 6
 AlFeature  = 1
 Width = 9000
 Height = 12000

=== Type of Article = 12

===ISSN[36072] Al Type = 10000
```

An R&D report of the
Communications Engineering Branch
Lister Hill National Center for Biomedical Communications
National Library of Medicine

September 2001

Contents

Organization of this report*. In this report we aim to present both our research in image analysis and understanding, as well as the practical considerations in creating an operational system. Following the background statement, project objectives, and project significance, Section 4 presents a system description, database design and organization, and testing to select the OCR system; Sections 5-7 describe the analysis and design of the key automated processes for zoning, labeling and reformatting; Section 8 describes lexical analysis work to improve productivity; Section 9 describes the design and implementation of operator workstations; Section 10 describes other systems designed to assist production control and supervision; Section 11 gives an account of performance evaluation; and Section 12 outlines next steps. References to the literature appear in Section 13.*

Automating the production of bibliographic records for MEDLINE

1. Background

The Communications Engineering Branch has had a longstanding involvement in document and biomedical imaging from many different standpoints: image capture, storage and retrieval, lossy and lossless compression, image enhancement and other types of image processing. Document imaging research, in particular, has been applied to the design and development of prototype systems to serve as testbeds for investigations into electronic archiving and preservation of library materials[1-4], automated interlibrary loan systems[5,6] , on-demand document delivery[7,8], and Internet-enabled document delivery and management for the end user[9-12] . In addition, we have engaged in several years of related and relevant R&D in document image analysis and understanding[13-19].

In 1996 following the unprecedented shutdown of the Federal Government, the National Library of Medicine unexpectedly lost its longstanding contractual arrangements for the keyboarding of bibliographic data into MEDLINE, with no immediate prospects for reinstating those contracts. A consequence of these events was that MEDLINE, our flagship database, was getting out of date at the rate of 30,000 to 40,000 citations every passing month. This clearly untenable situation proved to be an opportunity to improve upon the production performance of the labor intensive manual keyboarding of bibliographic data by combining key technologies already familiar to our R&D staff engaged in the projects listed above: in particular, document scanning, optical character recognition (OCR) and image analysis techniques[13,14,19].

To make an immediate difference we focused on the OCR conversion of the abstract, the largest chunk of a MEDLINE bibliographic record, amounting to a maximum of 400 words. In a few weeks we had created two workstations by integrating scanners and a low end OCR package already in our lab for NLM's Library Operations staff to use as a short term tool to try to tackle the growing backlog. It helped to some extent, but the error rate of the simple OCR package proved to be unacceptably high, requiring a great deal of manual correction.

Seeking a more satisfactory solution, we embarked on the design of a larger scale system centered on a multi-engine OCR system and operator workstations for scanning, keyboard entry and verification ("reconciling"). This first generation system[20] code-named *Medical Article Records System*, MARS-1, allowed a scan operator to capture the first page of every article and manually zone the article title and abstract in the TIFF image for conversion by OCR. An editor marked up the journal article with instructional notes for the keyboarders to enter other fields. Concurrently or at any time, a keyboarder entered into a template all fields except for the abstract. Since double-keying was considered necessary for high accuracy, a second keyboarder repeated this process for the same articles. These two manual entries (one from each keyboarder) were then compared by a DIFF module in a matching server, producing a "citation difference" file highlighting inconsistencies. A MATCH module then matched the article title field in the citation difference file with the article title from the OCR output. The system now "knew" that the abstract and the rest of the keyed-in fields belonged to a particular article, and the reconcile operators were presented with a combined set of fields to be verified and corrected, and then uploaded to the NLM mainframe for use by the library's indexers. The indexers added the appropriate descriptive information such as MeSH terms, thereby completing the bibliographic record to be added to MEDLINE.

At installation, in its skeletal form, MARS-1 produced about 67 completed records a day. Over the next several months, the system was scaled up to eventually double the number of machines and operators. This scaling up and, what was equally important, incremental improvements, increased the production rate to over 600 completed records a day, a third of the total requirement at NLM, the goal that had been set from the beginning. The goal was to produce a third by MARS, a third by contract keyboarders and the remaining third from SGML (later, XML) coded records sent to NLM directly by journal publishers.

While pursuing image analysis research toward the design of a more comprehensively automated system, we introduced incremental improvements to increase the productivity of MARS-1 operators as outlined below:

Barcode scanners. Finding the manual entry of the nine digit "MRI number" uniquely identifying the journal issue burdensome and error-prone, all workstations were equipped with a barcode scanner. The MRI is a very important number because it identifies the journal issue and is the gateway to all the articles in that issue, the image files and the OCR output files.

Click select special symbols. In the case of the Greek letters and biomedical symbols which are not recognized by the OCR, a window with a list of these symbols in words (ALPHA for α, for instance) was provided to enable the operators to click on the words which are then automatically entered in the right place in the text. So even though the OCR system does not recognize these symbols, we made it easier for the operators to enter them.

Programmed keys. At first diacritics, NIH grant numbers and databank accession numbers (e.g., from GenBank) were entered separately after the MARS process, turning out to be laborious. It was much better for the keyboarders to enter these as they encountered them in the article. So the keyboarding template software was modified to accommodate programmed keys for diacritics, resulting in a 4% improvement in production rate.

Spellcheck using biomedical lexicons. While OCR errors were expected, a more serious problem was the fact that the OCR system incorrectly assigned low confidence values to perfectly correct characters resulting in the highlighting of an excessive number of correct words on the workstation screen, requiring the reconcile operator to tab unnecessarily through all these correct words. The solution was to develop a spellcheck module relying on a combination of biomedical lexicons, the NLM's UMLS Metathesaurus and the Specialist Lexicon, and heuristic rules related to word lengths.[21] This feature cut down the highlighted correct words and the corresponding burden on the operators by about 50%. The net increase in production level due to this improvement was about 4%.

With MARS-1 in production, the design of the next generation system[20a] was begun to introduce more comprehensive automation and lower per-unit cost. MARS-2 is a database-centered and database-driven system that incorporates subsystems for automated page segmentation (zoning), automatic field identification (or labeling), automated syntax reformatting, and a classifier for the Greek letters and other symbols that the OCR system does not handle, all processes eliminating or reducing human labor. Three types of operators are still necessary: for scanning, editing and reconciling, but there are differences. The scan operator does less than he/she did in MARS-1 since manual zoning, a time consuming step, is eliminated. The "edit" operator combines the job of the editor and keyboarder in MARS-1, but much of the previous keyboarder's work is eliminated. The reconcile operator's task to verify and correct any errors remains the same. A description of MARS-2 appears in Section 4.

This report is organized as follows. Following the background statement, project objectives and project significance, Section 4 presents a system description including database organization and testing to select the OCR system; Sections 5-7 describe the image analysis and understanding work underlying the automated processes for zoning, labeling and reformatting; Section 8 describes lexical analysis techniques that improve the character and word recognition in the extracted fields; Section 9 describes the operator workstation design; Section 10 describes other systems designed to assist production control and supervision; Section 11 gives an account of performance evaluation; and Section 12 outlines next steps. References to the literature appear at the end.

2. Project objectives

The objectives of this project are to:

(a) research and apply document image analysis and understanding techniques to the problem of automating the production of bibliographic records for MEDLINE;

(b) build, operate and maintain a practical production system for this purpose;

(c) use the production system as an experimental testbed to conduct research in document image analysis and understanding techniques, to identify opportunities for improved performance, and to optimize the performance of manual processes inevitable in a practical system (e.g., verification of the output of automated processes);

(d) redesign and modify subsystems, or create new subsystems, to achieve improved performance;

(e) enable the computer science and informatics research communities to conduct further image analysis research toward the development of algorithms, tools and techniques for automated data extraction and other applications by providing ground truth data.

3. Project significance

There are two principal and obvious reasons why automated data entry is of interest: first, the gradual rise of labor costs; and second, the unrelenting increase in the amount of data that needs to be entered into databases from paper-based information. The vast majority of the hundreds of databases produced in every discipline rely on laborious keyboard entry of bibliographic information from articles in journals, e.g., article title, author names, institutions, abstract, dates, page numbers, etc. Image analysis and understanding techniques provide the basis for the development of automated systems that promise a cheaper alternative to keyboarding, and a more timely availability of bibliographic data for the public.

4. System description

To provide context for the subsequent discussions of image analysis research, in this section we first describe the overall system (MARS-2) that serves both as an experimental testbed as well as a practical production tool. Secondly, since the system is database-centered and database-driven, we outline the design of the central database that controls the workflow and which serves as the repository for all data flowing in and out of the many processes. Thirdly, since the optical character recognition system is key to the extraction of text from the document images, we describe the evaluation criteria and test results that pointed to the package selected.

4.1 Overview

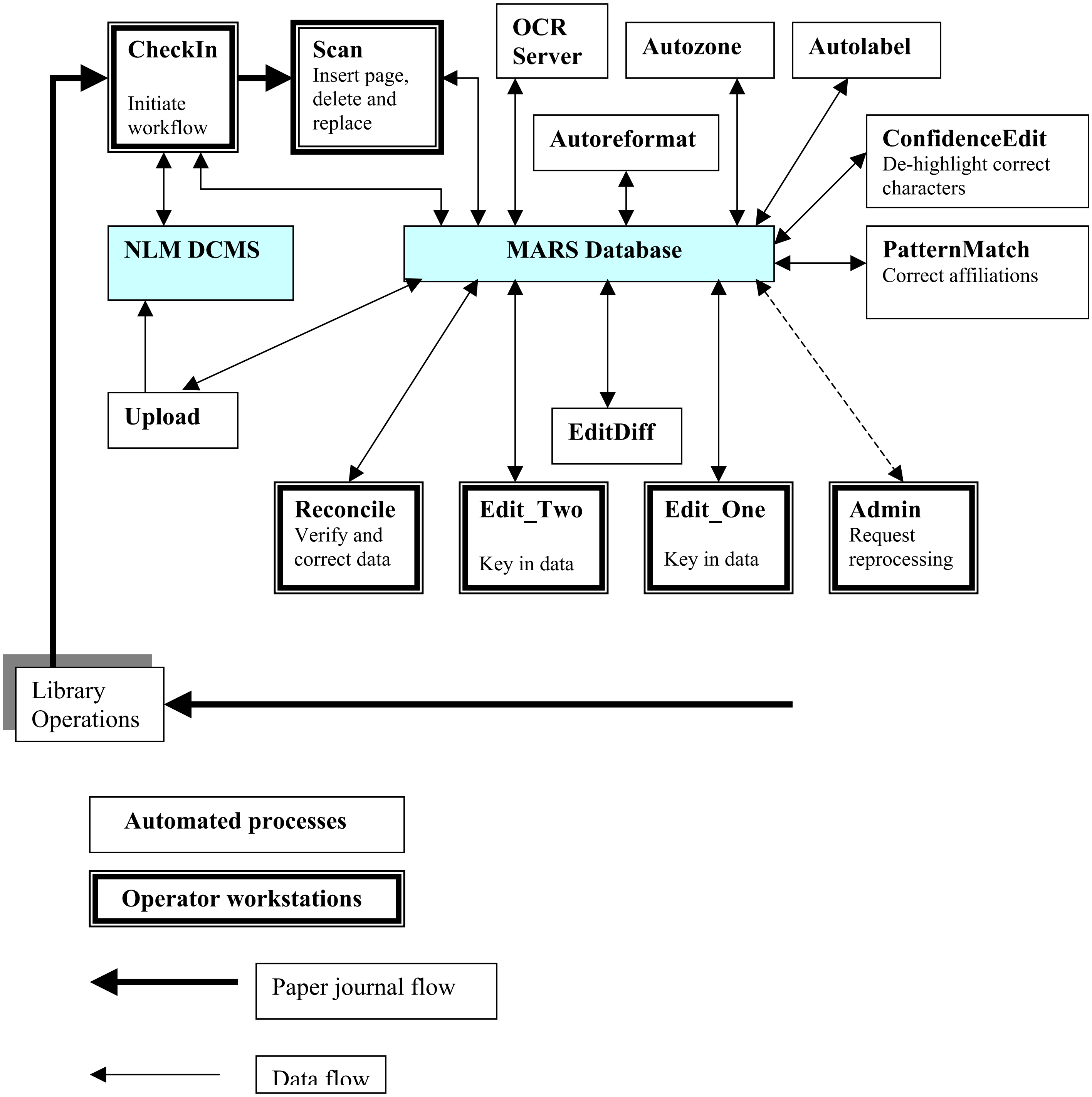
The MARS-2 system consists of both automated and operator-controlled subsystems as shown in Figure 4.1. The schematic shows automated processes as boxes with thin boundaries, and manual workstations with thick boundaries. The workflow is initiated at the CheckIn stage where a supervisor scans the barcode on a journal issue arriving at the production facility. As mentioned earlier, this barcode number, called the “MRI”, is routinely affixed to every journal issue by NLM staff. It therefore serves as a unique key to identify the issue, all the pages scanned in that issue, and indeed the outputs of all processes performed on those page images. The scanning operator captures the first page of every article in the issue, since this page contains the fields we seek to extract automatically. The resulting TIFF images go into a file server and associated data into the MARS database for which the underlying DBMS is Microsoft’s SQL Server. The OCR system accesses the TIFF images and produces the corresponding text as well as other data descriptive of the text characters such as bounding boxes, attributes (bold, italic, underlined), confidence level, font style and size, and others. The automatic zoning (Autozone) module then blocks out the contiguous text using features derived from the OCR output data, followed by the automated labeling (Autolabel) module that identifies the zones as the fields of interest (article title, author names, affiliations, abstract). The Autoreformat module then organizes the syntax of the zone contents to adhere to MEDLINE conventions (e.g., author name *John A. Smith* becomes *Smith JA*).

At this point, two lexicon-enabled modules operate on the data to reduce the burden on the operator performing the final checking and verification of the data: ConfidenceEdit modifies the incorrect confidence levels assigned to the characters by the OCR system, and PatternMatch corrects institutional affiliations whose text is frequently recognized incorrectly by the OCR system.

Some data cannot be automatically extracted. The major reason is that they appear in pages other than the scanned first page. Such data is manually entered by a pair of edit operators, a double-key process that ensures a high degree of accuracy. An EditDiff module then correlates these different entries and notes differences. The output of the automated processes and the edit operators is then presented to the reconcile operator who verifies and corrects the text. The Upload module then sends the verified data to the NLM's DCMS (Data Creation Maintenance System) which is accessed by NLM indexers to add MeSH terms and keywords, thereby completing the MEDLINE record.

The Admin workstation shown is used by the production supervisor to send a journal issue back to an earlier processing stage in case of errors.

Figure 4.1 MARS-2 general schematic



4.2 Database considerations

The MARS-2 system, as shown in Figure 4.1, is database-centered, and all data pertaining to its operation is stored in an RDBMS, entered by certain processes and retrieved by others. The RDBMS chosen for the MARS database is Microsoft SQL Server 6.5 running under the Windows NT 4.0 Server operating system. This RDBMS is scheduled for a major upgrade to MS SQL Server 2000 Enterprise Edition under Windows 2000 Advanced Server with the Clustering / High Availability functionality. This upgrade will provide the following advantages: a row-level locking scheme that minimizes deadlocks, cascading referential integrity constraints that simplify the implementation of business logic in applications, and functionality that enables XML to be used to retrieve and modify values in the database. Other features of this upgraded system are its availability of data types to handle Unicode (if this is considered useful in the future), and user-friendly GUI-based tools for easier maintenance of the database system.

The database-driven workflow in MARS is controlled by a module called the *Work Distribution Manager (WDM)* that uses a set of database tables to determine when processes (e.g., automated zoning, labeling, reformatting, etc.) should act on image and text data, when the processes are completed, and which process must occur next in the workflow. WDM resides within the database system, and relies on stored procedures and scheduler to achieve its functionality.

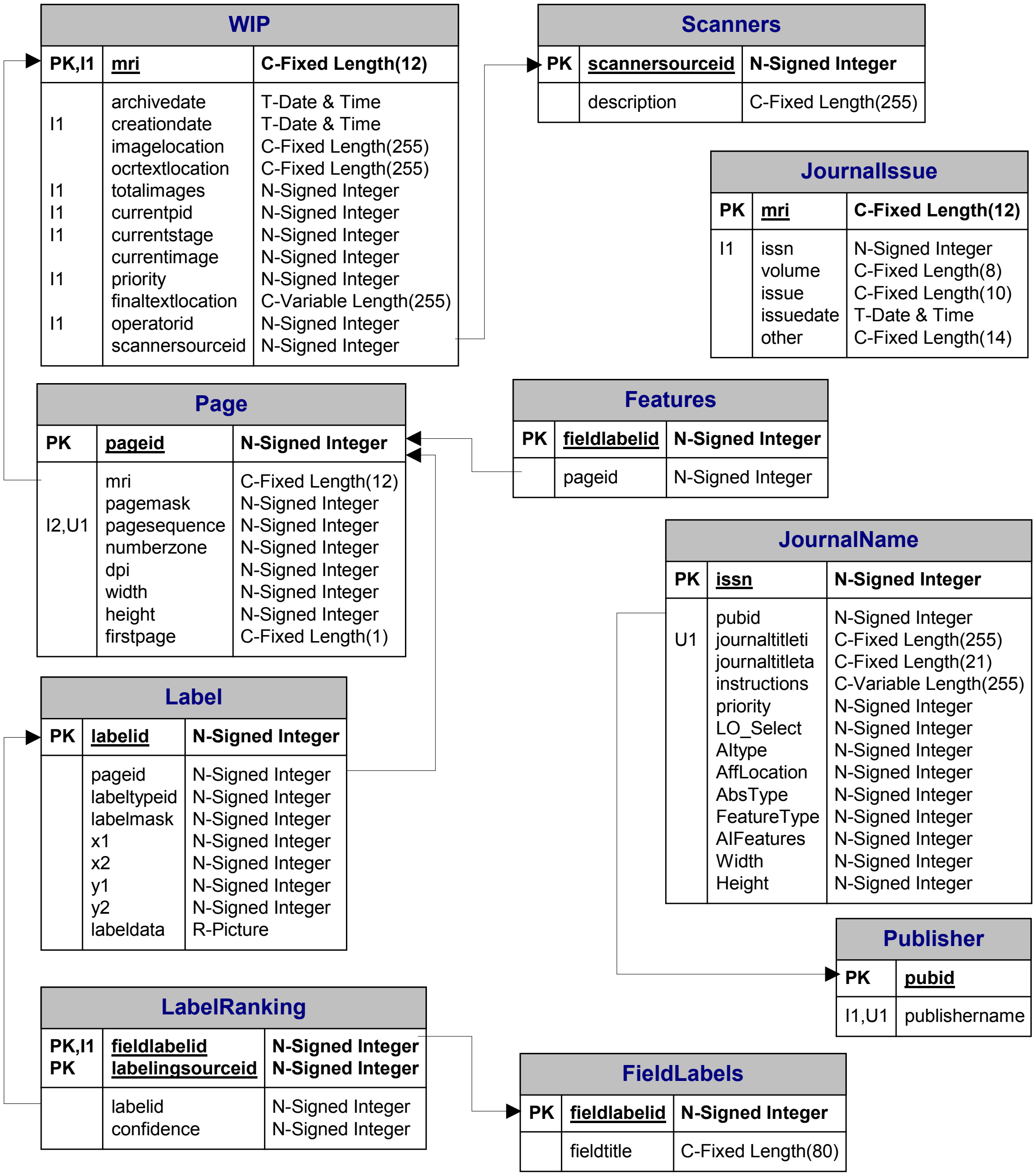
The MARS database consists of sixty two tables, some of which are outlined in this section and shown in the figures. The *WorkInProgress* table (WIP) serves as a central hub containing common information for each database record (set of data related to a single journal issue) and keeps track of whether the processing of data for a particular journal issue is incomplete, has been completed, or is waiting to be archived. WIP uses the MRI (journal issue identification number) as a primary key, and contains such information as: the date the record is created, the date it is uploaded, location of the page images and the text, total number of page images scanned, current process in progress (by the PID or process identification number), the current stage of the process, the priority of the journal as established by NLM’s Library Operations, and other data.

The *Page Table* contains data associated with a single scanned page. It keeps one level of a tree of relationships in which a journal issue has many page images, a page has several zones, zones have many text lines, and text lines have characters. A row in this table is created by the Scan process, but most of the information is filled in by the output of the OCR. It also contains information on the width and height of a page in 300 dpi units.

The *Label Table* defines all the zones, both from the scanner and from the OCR system. For most records it will reflect OCR related information at the zone level, such as zone sequence and zone coordinates, and also provide a link to the text output from the OCR. Each process creates only its own labels instead of modifying labels that were created by other processes, to preserve the historical data for journal issues processed through the system.

The *Field Label Table* defines all the fields that identify a citation, viz., article title, author, affiliation and abstract label among other labels.

MARS II Database System
Major Tables



WIP

PK,I1	mri	C-Fixed Length(12)
	archivedate	T-Date & Time
I1	creationdate	T-Date & Time
	imagelocation	C-Fixed Length(255)
	ocrtextlocation	C-Fixed Length(255)
I1	totalimages	N-Signed Integer
I1	currentpid	N-Signed Integer
I1	currentstage	N-Signed Integer
	currentimage	N-Signed Integer
I1	priority	N-Signed Integer
	finaltextlocation	C-Variable Length(255)
I1	operatorid	N-Signed Integer
	scannersourceid	N-Signed Integer

Scanners

PK	scannersourceid	N-Signed Integer
	description	C-Fixed Length(255)

JournalIssue

PK	mri	C-Fixed Length(12)
I1	issn	N-Signed Integer
	volume	C-Fixed Length(8)
	issue	C-Fixed Length(10)
	issuedate	T-Date & Time
	other	C-Fixed Length(14)

Page

PK	pageid	N-Signed Integer
	mri	C-Fixed Length(12)
	pagemask	N-Signed Integer
I2,U1	pagesequence	N-Signed Integer
	numberzone	N-Signed Integer
	dpi	N-Signed Integer
	width	N-Signed Integer
	height	N-Signed Integer
	firstpage	C-Fixed Length(1)

Features

PK	fieldlabelid	N-Signed Integer
	pageid	N-Signed Integer

JournalName

PK	issn	N-Signed Integer
	pubid	N-Signed Integer
U1	journaltitleti	C-Fixed Length(255)
	journaltitleta	C-Fixed Length(21)
	instructions	C-Variable Length(255)
	priority	N-Signed Integer
	LO_Select	N-Signed Integer
	AItype	N-Signed Integer
	AffLocation	N-Signed Integer
	AbsType	N-Signed Integer
	FeatureType	N-Signed Integer
	AIFeatures	N-Signed Integer
	Width	N-Signed Integer
	Height	N-Signed Integer

Label

PK	labelid	N-Signed Integer
	pageid	N-Signed Integer
	labeltypeid	N-Signed Integer
	labelmask	N-Signed Integer
	x1	N-Signed Integer
	x2	N-Signed Integer
	y1	N-Signed Integer
	y2	N-Signed Integer
	labeldata	R-Picture

Publisher

PK	pubid
I1,U1	publishername

LabelRanking

PK,I1 PK	fieldlabelid labelingsourceid	N-Signed Integer N-Signed Integer
	labelid	N-Signed Integer
	confidence	N-Signed Integer

FieldLabels

PK	fieldlabelid	N-Signed Integer
	fieldtitle	C-Fixed Length(80)

Figure 4.2.1

A complete listing of all the rules required for automated processing appears in the *Rules Table.* There are two types of rules: edit rules and format rules. Edit rules are for special processing to offset OCR errors in the context of a word. For example, a zero (0) occurring in the name “F0rd” would be reset to an "o". The more numerous format rules, performed in a series of loops, implement the reformatting of field syntax, e.g., author names appearing in a journal article in the conventional first name-middle initial-last name sequence would be reformatted to the last name-space-first and middle initials form required by MEDLINE. This table contains an identification number for each rule, the text string that triggers a rule, and a text string that the rule triggers. This is discussed further in Section 7 on Automated Reformatting.

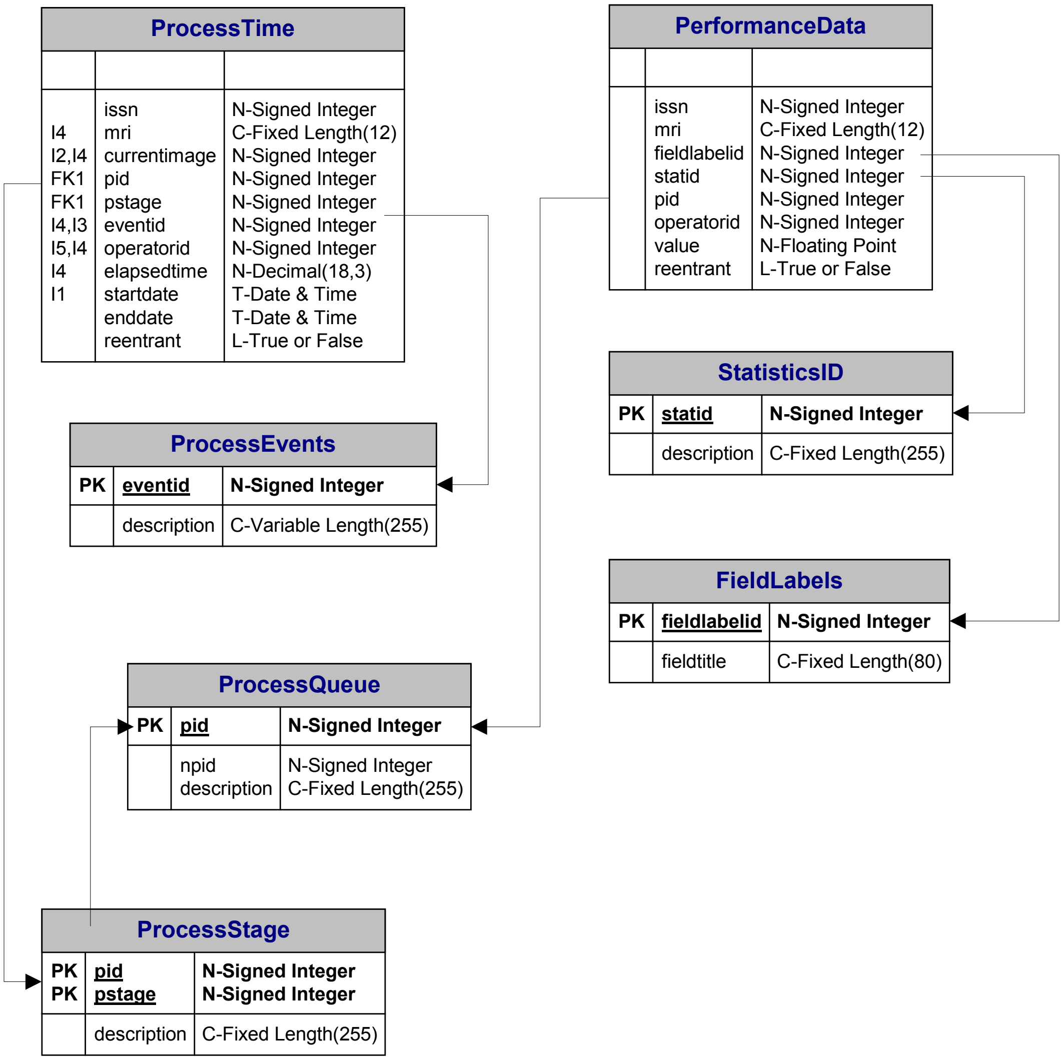
MARS II Database System
Format Rules

Rules		
PK	**ruleid**	**N-Signed Integer**
	descriptionid matchstring actionstring	N-Signed Integer C-Variable Length(255) C-Variable Length(255)

LookupRules		
PK **PK** **PK**	**ruleid** **fieldlabelid** **issn**	**N-Signed Integer** **N-Signed Integer** **N-Signed Integer**

Other tables keep track of performance data (time taken for a process to complete, number of text lines written per second, number of words spell checked, etc.), scanner settings, operator names and their work statistics, list of processes, the current stage a process is in, and errors that are tracked. Stored procedures were created to support reports of performance statistics. These tables are shown in the figure titled Instrumentation, below.

MARS II Database System
Instrumentation



ProcessTime

	issn	N-Signed Integer
I4	mri	C-Fixed Length(12)
I2,I4	currentimage	N-Signed Integer
FK1	pid	N-Signed Integer
FK1	pstage	N-Signed Integer
I4,I3	eventid	N-Signed Integer
I5,I4	operatorid	N-Signed Integer
I4	elapsedtime	N-Decimal(18,3)
I1	startdate	T-Date & Time
	enddate	T-Date & Time
	reentrant	L-True or False

PerformanceData

	issn	N-Signed Integer
	mri	C-Fixed Length(12)
	fieldlabelid	N-Signed Integer
	statid	N-Signed Integer
	pid	N-Signed Integer
	operatorid	N-Signed Integer
	value	N-Floating Point
	reentrant	L-True or False

StatisticsID

PK	**statid**	**N-Signed Integer**
	description	C-Fixed Length(255)

ProcessEvents

PK	**eventid**	**N-Signed Integer**
	description	C-Variable Length(255)

FieldLabels

PK	**fieldlabelid**	**N-Signed Integer**
	fieldtitle	C-Fixed Length(80)

ProcessQueue

PK	**pid**	**N-Signed Integer**
	npid	N-Signed Integer
	description	C-Fixed Length(255)

ProcessStage

PK	**pid**	**N-Signed Integer**
PK	**pstage**	**N-Signed Integer**
	description	C-Fixed Length(255)

4.3 OCR system evaluation and selection

A key step in the design of MARS is the selection of the OCR system. This selection was based on a performance comparison of six commercial packages, four of them single-engine systems and Maxsoft-Ocron and Prime Recognition, both multiple-engine voting systems. Testing was done on about 20,000 characters from 15 page images (scanned at 300 dpi) from five biomedical journals indexed in MEDLINE. All five journals were selected because they appeared likely to cause conversion problems, e.g., because of tightly packed text and small sized fonts.

The testing focused primarily on the number of error blocks (a "block" may be either a character or a word, depending on the OCR package) in line with the following error criteria: (1) highlighted correct blocks, i.e., blocks that are highlighted by the OCR system, but whose contents are correct (**false alarms**); (2) highlighted error blocks, blocks that are highlighted, and whose contents have incorrect characters and correction is required (**correctly detected errors**); (3) undetected blocks, which have incorrect characters that are not highlighted (**undetected errors**). The data below shows Prime Recognition superior with respect to all three error criteria.

False alarms	**Correctly detected errors**	**Undetected errors**
Prime Recognition = *168*	Prime Recognition = *42*	Prime Recognition = *6*
Wordscan = 339	Wordscan = 72	Wordscan = 24
TextBridge = 70	TextBridge = 70	TextBridge = 37
Omnipage = 285	Omnipage = 66	Omnipage = 23
Cuneiform = 259	Cuneiform = 53	Cuneiform = 35
Maxsoft-Ocron = n/a	Maxsoft-Ocron = n/a	Maxsoft-Ocron = 33

Other evaluation factors, important in a practical production system, included: (1) capability to proofread with a displayed bitmap, (2) medical dictionary interface, and (3) accessibility from our application software. Only Prime Recognition fully met all the criteria, the limitations of the other packages noted below.

Proofing with bitmap	**Medical dictionary interface**	**Application-accessibility**
Prime Recognition= Yes	Prime Recognition = Yes	Prime Recognition = Yes
Wordscan= small bitmap	Wordscan= 13 char. limit	Wordscan = Yes
TextBridge=need Word/ WP	TextBridge=10K word limit	TextBridge = No
Omnipage= Yes	Omnipage= 5K word limit	Omnipage = Yes
Cuneiform= Yes	Cuneiform= 24 char. limit	Cuneiform = No
Maxsoft-Ocron= No	Maxsoft-Ocron= Yes	Maxsoft-Ocron = Yes

In addition to character codes, the Prime Recognition OCR, in its output, provides rich secondary data, e.g., character coordinates, confidence levels, font size, font attributes and many others, much of which is exploited by downstream processes, as described later in this report.

To incorporate the OCR software into the MARS system, we developed a module, Prime Recognition OCR Daemon (PROD) that consists of a C++ class that acts as a wrapper for the Prime Recognition C API, and also communicates with the MARS database. In addition, it incorporates two other modules: (a) Bounding Box Corrector, software library routines developed in cooperation with scientists at MathSoft, Inc.; and (b) an independent OCR package from ScanSoft. The first is needed to correct the character coordinates from the Prime Recognition OCR system to improve the reliability of our inhouse zone correction algorithm (Section 5). The second provides more reliable initial segmentation than the engines in the Prime Recognition package do for certain journal layout types. For these journals, identified by ISSN in the database, the zones from ScanSoft are used as a starting point for our zone correction algorithm.

PROD is designed for flexibility. For example, we can set it for the number of CPUs active in the OCR server, the number of OCR engines, for the correction of character coordinates and for recording the time duration for OCR processing. Also, PROD may be used to poll the database for journal issues ready to be processed by the OCR, and to begin or stop processing.

5. Automated Zoning

The first step after the conversion of the bitmapped image by the OCR system is to apply image analysis techniques to block out (“zone”) the regions of contiguous text, in particular those text groups corresponding to the bibliographic fields of highest interest: viz., article title, authors, affiliation, and abstract. Our survey of ongoing research in the application of image analysis to automated zoning described in the literature appears in Section 5.1.

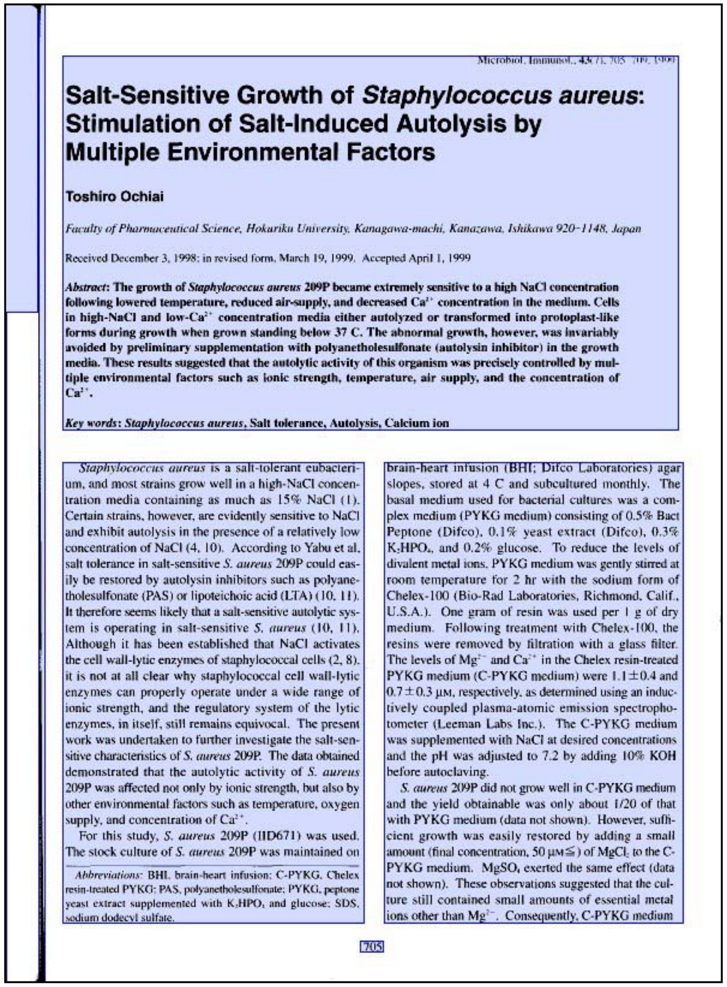
The commercial OCR system used in the MARS system includes a package to perform automatic zoning. However, our experiments showed that this function does not segment the page images into zones containing the bibliographic fields of interest with sufficient accuracy. The most common error made by the commercial automatic zoning function is that zones are too large and include more than one significant text group. Figure 5.1 illustrates a typical case where the title, author, abstract and affiliation are all in one zone, along with extraneous publication identification. Figure 5.2 illustrates another case where, in addition to the previous problem, a two-column abstract is grouped inappropriately into a single zone. In this example, the text lines in the two columns are joined, disrupting the proper reading order. For example, the middle text of the first line of the zone is incorrectly read as "..models have opment of..."

Correct zones are critical to downstream processes in MARS-2. The stage that follows automated zoning is the automated labeling of the zones as title, authors, affiliation and abstract.[22] This complex labeling process uses several items of information in each zone to determine its identity. Information used to label a zone include absolute and relative location of the zone, and key words within the zone. Clearly, the zone region must be correct if it is to provide useful information to the labeling program.

Downstream from automatic zoning and labeling, the title, author and affiliation fields are automatically reformatted to comply with MEDLINE conventions[23]. This process also depends on correctly sized and labeled zones to be effective. Incorrect zones confound reformatting, ultimately requiring time-consuming manual intervention at the reconcile stage, thereby offsetting the advantage expected from an automated system.

An alternative to automatic zoning is to require operators to manually draw, using special software and the mouse, correct zones onto the bitmapped images prior to the OCR process. This was done in the MARS-1 system to identify the title and abstract zones. It took operators about 14 seconds per image to draw these two zones. For the four zones needed in MARS-2, we can estimate that it would require about 28 seconds per image of operator time to perform manual zoning, a considerable burden. For example, for a production rate of 1,000 records a day, manual zoning would add over 7 person hours of labor, approximately equivalent to an additional full time worker.

Since we cannot depend on the commercial OCR system to correctly zone images, and seek to eliminate manual zoning, we developed our own automatic zoning capability. With our own process, we free ourselves from depending on the commercial OCR system for automatic zoning, and can tailor the zone program design and operating parameters for images from the specific biomedical journals relevant to MEDLINE. However, rather than starting from scratch, we combine the automated zoning capability of the OCR system with our added functionality for zone correction.



Microbiol. Immunol., 43(7), 705–710, 1999

Salt-Sensitive Growth of *Staphylococcus aureus*: Stimulation of Salt-Induced Autolysis by Multiple Environmental Factors

Toshiro Ochiai

Faculty of Pharmaceutical Science, Hokuriku University, Kanagawa-machi, Kanazawa, Ishikawa 920–1148, Japan

Received December 3, 1998; in revised form, March 19, 1999. Accepted April 1, 1999

***Abstract*: The growth of *Staphylococcus aureus* 209P became extremely sensitive to a high NaCl concentration following lowered temperature, reduced air-supply, and decreased Ca^{2+} concentration in the medium. Cells in high-NaCl and low-Ca^{2+} concentration media either autolyzed or transformed into protoplast-like forms during growth when grown standing below 37 C. The abnormal growth, however, was invariably avoided by preliminary supplementation with polyanetholesulfonate (autolysin inhibitor) in the growth media. These results suggested that the autolytic activity of this organism was precisely controlled by multiple environmental factors such as ionic strength, temperature, air supply, and the concentration of Ca^{2+}.**

***Key words*: *Staphylococcus aureus*, Salt tolerance, Autolysis, Calcium ion**

Staphylococcus aureus is a salt-tolerant eubacterium, and most strains grow well in a high-NaCl concentration media containing as much as 15% NaCl (1). Certain strains, however, are evidently sensitive to NaCl and exhibit autolysis in the presence of a relatively low concentration of NaCl (4, 10). According to Yabu et al, salt tolerance in salt-sensitive *S. aureus* 209P could easily be restored by autolysin inhibitors such as polyanetholesulfonate (PAS) or lipoteichoic acid (LTA) (10, 11). It therefore seems likely that a salt-sensitive autolytic system is operating in salt-sensitive *S. aureus* (10, 11). Although it has been established that NaCl activates the cell wall-lytic enzymes of staphylococcal cells (2, 8). it is not at all clear why staphylococcal cell wall-lytic enzymes can properly operate under a wide range of ionic strength, and the regulatory system of the lytic enzymes, in itself, still remains equivocal. The present work was undertaken to further investigate the salt-sensitive characteristics of *S. aureus* 209P. The data obtained demonstrated that the autolytic activity of *S. aureus* 209P was affected not only by ionic strength, but also by other environmental factors such as temperature, oxygen supply, and concentration of Ca^{2+}.

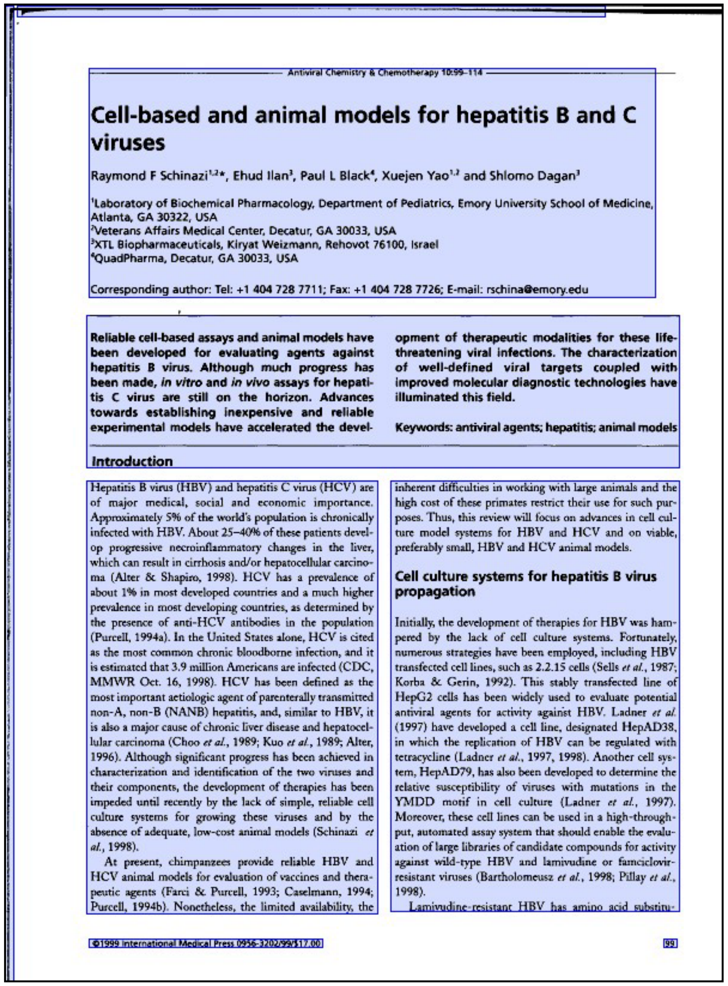
For this study, *S. aureus* 209P (IID671) was used. The stock culture of *S. aureus* 209P was maintained on brain-heart infusion (BHI; Difco Laboratories) agar slopes, stored at 4 C and subcultured monthly. The basal medium used for bacterial cultures was a complex medium (PYKG medium) consisting of 0.5% Bact Peptone (Difco), 0.1% yeast extract (Difco), 0.3% K$_2$HPO$_4$, and 0.2% glucose. To reduce the levels of divalent metal ions, PYKG medium was gently stirred at room temperature for 2 hr with the sodium form of Chelex-100 (Bio-Rad Laboratories, Richmond, Calif., U.S.A.). One gram of resin was used per 1 g of dry medium. Following treatment with Chelex-100, the resins were removed by filtration with a glass filter. The levels of Mg^{2+} and Ca^{2+} in the Chelex resin-treated PYKG medium (C-PYKG medium) were 1.1±0.4 and 0.7±0.3 μM, respectively, as determined using an inductively coupled plasma-atomic emission spectrophotometer (Leeman Labs Inc.). The C-PYKG medium was supplemented with NaCl at desired concentrations and the pH was adjusted to 7.2 by adding 10% KOH before autoclaving.

S. aureus 209P did not grow well in C-PYKG medium and the yield obtainable was only about 1/20 of that with PYKG medium (data not shown). However, sufficient growth was easily restored by adding a small amount (final concentration, 50 μM≦) of MgCl$_2$ to the C-PYKG medium. MgSO$_4$ exerted the same effect (data not shown). These observations suggested that the culture still contained small amounts of essential metal ions other than Mg^{2+}. Consequently, C-PYKG medium

Abbreviations: BHI, brain-heart infusion; C-PYKG, Chelex resin-treated PYKG; PAS, polyanetholesulfonate; PYKG, peptone yeast extract supplemented with K$_2$HPO$_4$ and glucose; SDS, sodium dodecyl sulfate.

705

Figure 5.1 An example of large zones generated by the commercial OCR system.



Antiviral Chemistry & Chemotherapy 10:99–114

Cell-based and animal models for hepatitis B and C viruses

Raymond F Schinazi[1,2]*, Ehud Ilan[3], Paul L Black[4], Xuejen Yao[1,2] and Shlomo Dagan[3]

[1]Laboratory of Biochemical Pharmacology, Department of Pediatrics, Emory University School of Medicine, Atlanta, GA 30322, USA
[2]Veterans Affairs Medical Center, Decatur, GA 30033, USA
[3]XTL Biopharmaceuticals, Kiryat Weizmann, Rehovot 76100, Israel
[4]QuadPharma, Decatur, GA 30033, USA

Corresponding author: Tel: +1 404 728 7711; Fax: +1 404 728 7726; E-mail: rschina@emory.edu

Reliable cell-based assays and animal models have been developed for evaluating agents against hepatitis B virus. Although much progress has been made, *in vitro* and *in vivo* assays for hepatitis C virus are still on the horizon. Advances towards establishing inexpensive and reliable experimental models have accelerated the development of therapeutic modalities for these life-threatening viral infections. The characterization of well-defined viral targets coupled with improved molecular diagnostic technologies have illuminated this field.

Keywords: antiviral agents; hepatitis; animal models

Introduction

Hepatitis B virus (HBV) and hepatitis C virus (HCV) are of major medical, social and economic importance. Approximately 5% of the world's population is chronically infected with HBV. About 25–40% of these patients develop progressive necroinflammatory changes in the liver, which can result in cirrhosis and/or hepatocellular carcinoma (Alter & Shapiro, 1998). HCV has a prevalence of about 1% in most developed countries and a much higher prevalence in most developing countries, as determined by the presence of anti-HCV antibodies in the population (Purcell, 1994a). In the United States alone, HCV is cited as the most common chronic bloodborne infection, and it is estimated that 3.9 million Americans are infected (CDC, MMWR Oct. 16, 1998). HCV has been defined as the most important aetiologic agent of parenterally transmitted non-A, non-B (NANB) hepatitis, and, similar to HBV, it is also a major cause of chronic liver disease and hepatocellular carcinoma (Choo *et al.*, 1989; Kuo *et al.*, 1989; Alter, 1996). Although significant progress has been achieved in characterization and identification of the two viruses and their components, the development of therapies has been impeded until recently by the lack of simple, reliable cell culture systems for growing these viruses and by the absence of adequate, low-cost animal models (Schinazi *et al.*, 1998).

At present, chimpanzees provide reliable HBV and HCV animal models for evaluation of vaccines and therapeutic agents (Farci & Purcell, 1993; Caselmann, 1994; Purcell, 1994b). Nonetheless, the limited availability, the inherent difficulties in working with large animals and the high cost of these primates restrict their use for such purposes. Thus, this review will focus on advances in cell culture model systems for HBV and HCV and on viable, preferably small, HBV and HCV animal models.

Cell culture systems for hepatitis B virus propagation

Initially, the development of therapies for HBV was hampered by the lack of cell culture systems. Fortunately, numerous strategies have been employed, including HBV transfected cell lines, such as 2.2.15 cells (Sells *et al.*, 1987; Korba & Gerin, 1992). This stably transfected line of HepG2 cells has been widely used to evaluate potential antiviral agents for activity against HBV. Ladner *et al.* (1997) have developed a cell line, designated HepAD38, in which the replication of HBV can be regulated with tetracycline (Ladner *et al.*, 1997, 1998). Another cell system, HepAD79, has also been developed to determine the relative susceptibility of viruses with mutations in the YMDD motif in cell culture (Ladner *et al.*, 1997). Moreover, these cell lines can be used in a high-throughput, automated assay system that should enable the evaluation of large libraries of candidate compounds for activity against wild-type HBV and lamivudine or famciclovir-resistant viruses (Bartholomeusz *et al.*, 1998; Pillay *et al.*, 1998).

Lamivudine-resistant HBV has amino acid substitu-

©1999 International Medical Press 0956-3202/99/$17.00

99

Figure 5.2 A second example of large zones generated by the commercial OCR system.

5.1 Methods and Procedures

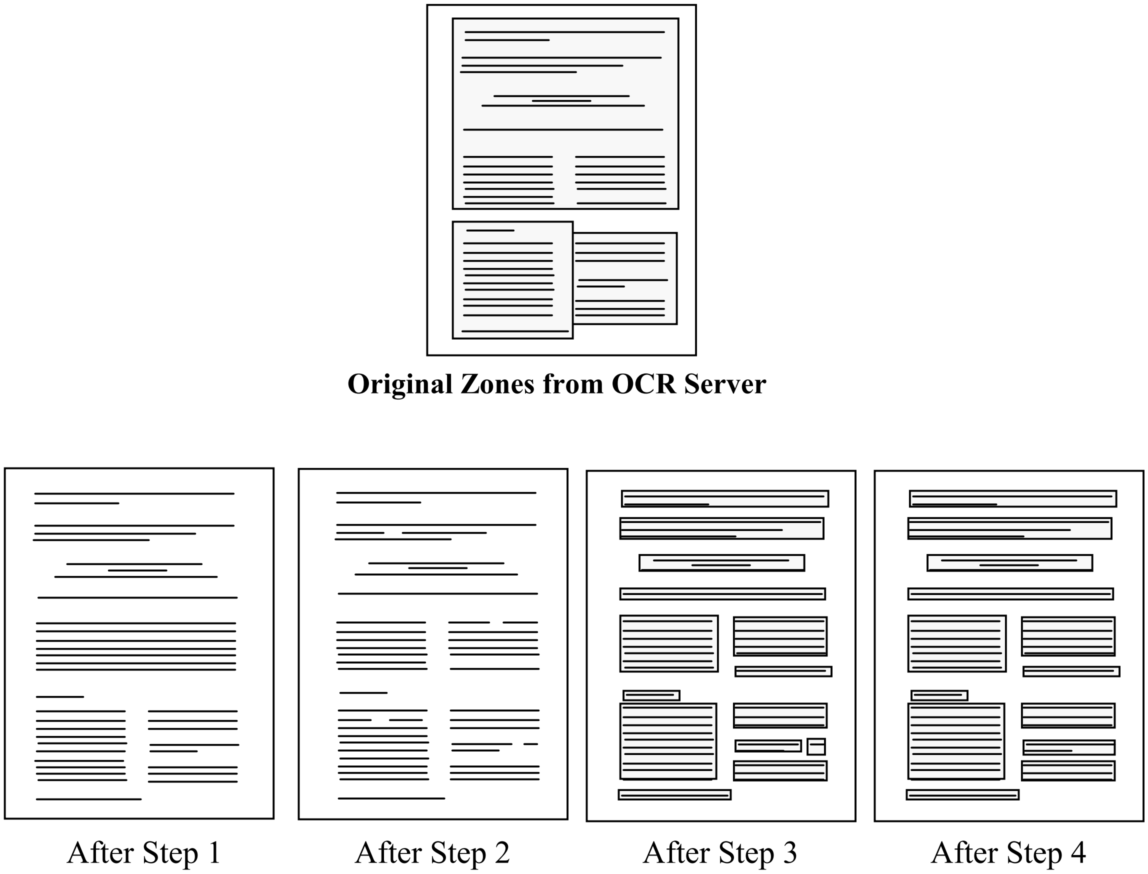
Much of the research reported in the literature employ methods analogous to those used to isolate and separate characters (symbol isolation) to segment page images into zones. A brief survey of activity in automatic zoning methods is given in Jain[24]. Approaches include "top-down" methods[25], which segment a page by x-cuts and y-cuts into smaller regions, "bottom-up"[26,27], which recursively grow homogeneous regions from small components, and combinations of both[24,28]. Tradeoff factors include: granularity (finding small enough zones), computation time, and sensitivity to input parameters such as noise, skew and page orientation[29-31]. Top-down methods tend to be faster and less sensitive to input parameters and page orientation, but require pages to have a "Manhattan layout", i.e., the blocks may be separated by vertical and horizontal lines. Bottom-up and combination methods often result in greater accuracy at the expense of computational complexity and sensitivity to input parameters. All of these methods zone the page using image data alone, prior to OCR conversion. Since the reported performance is variable, and because rich secondary data is available from our OCR system, our approach, in contrast, is to exploit the output data of the OCR system to implement automatic zoning.

As noted earlier, in addition to ASCII text, the OCR system provides information about each of the converted characters in the output file. This information includes the level of confidence that the character was correctly recognized, character attributes such as italic or bold, character point size, and the x and y coordinates of the rectangle that bounds the character (bounding boxes)[32]. Thus we have both geometric and non-geometric feature information available for each converted character. Our approach is to draw upon these features to group text into correct zones. For example, we use the bounding box coordinates to determine which characters are grouped closely in the same region on the page. Information on character size and attributes provide additional clues for keeping groups of adjacent characters together or placing them in separate zones.

Our zone correction method uses both top-down and bottom-up design strategies[41], used by other investigators on image data, on our OCR output (non-image) data. The overall procedure is outlined in Table 5.1, and an example is given in Figure 5.3.

Table 5.1 Zone correction program steps

	Input	Function	Output
1.	Zones and data from OCR system	Separate zones into text lines	Text lines
2.	Text lines	Separate lines into fragments	Text lines, fragments
3.	Lines and line fragments	Combine lines vertically into zones	Initial zones
4.	Initial zones	Combine zones horizontally into zones	Final zones



Original Zones from OCR Server

After Step 1 After Step 2 After Step 3 After Step 4

Figure 5.3 Zone correction program steps

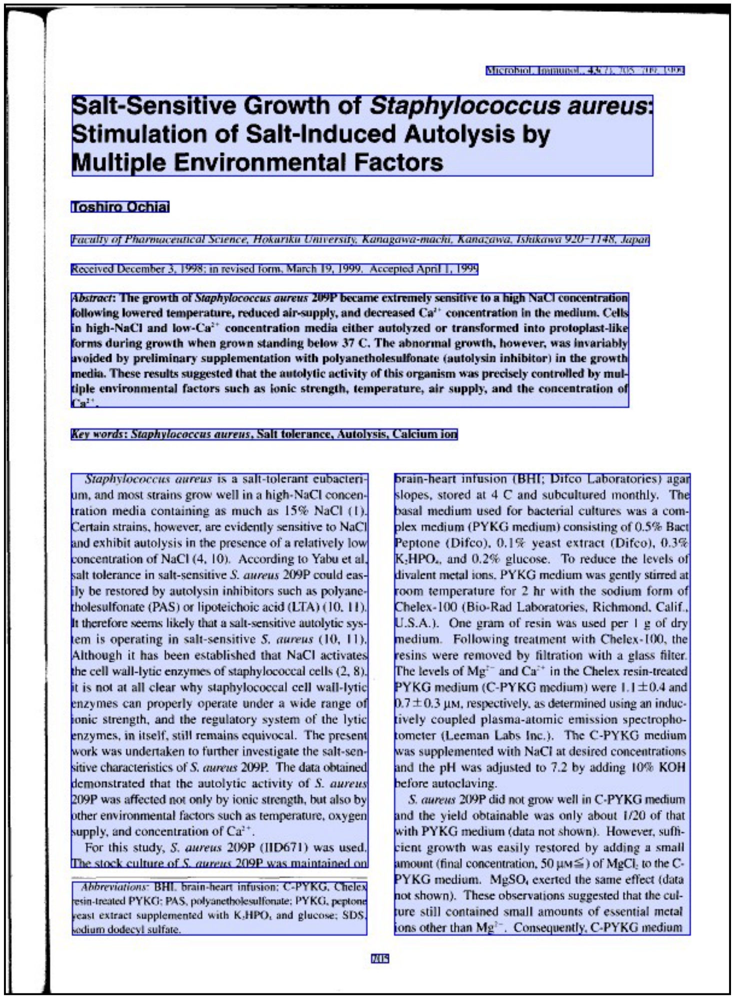
The first step in creating new zones is to disassemble the original zones from the OCR system. Each zone is divided into individual text lines. In step 2, lines are further split horizontally into multiple lines when the space between words exceeds a distance threshold (empirically determined). This occasionally results in unnecessarily splitting lines into multiple parts, but is needed in order to split lines that originally span across two closely spaced columns, as shown in Figure 5.3. Some of these lines will be rejoined in later steps. The bounding box enclosing each line is computed, as are several features such as percent italic characters and average character height. Some character features, such as bold or italic, are available directly from the OCR output data. Others, such as character height or case (upper or lower), are computed from the OCR output data.

In step 3, we combine the lines vertically into initial zones. The criteria for combining are that (a) the vertical distance between lines must be less than a threshold (again, empirically determined); (b) either the left edge, right edge or midpoint must be horizontally aligned; and (c) the features computed in the previous step must be similar. When a line is added to a zone, the zone's rectangular boundary is expanded to include the new line. Then all remaining lines are checked to see if they fall within the new zone.

If so, they are added to the zone. Many of the horizontally split lines are recombined in this way.

On rare occasion, some zones created in step 3 are too narrow. In this event, the fourth and last step is to combine such zones horizontally using criteria similar to those in the previous step. Here, the initial zones are combined if (a) the horizontal distance between the zones is less than a threshold; (b) either the top or bottom edges of the zones are vertically aligned; and (c) the computed features of the two zones are similar. When zones are thus merged, a new zone boundary rectangle is created to include both zones. Any smaller zones that fall within the rectangle are subsumed within this zone.

Figures 5.4 and 5.5 show the results of these steps applied to the two images used as examples in Figures 5.1 and 5.2. In both of these images, the title, author, affiliation and abstract are enclosed in separate zones, as required. In addition, in Figure 5.5, the two columns of the abstract are in separate zones. These two zones will be identified as abstract by the automated labeling process, which follows the zone correction process, and the enclosed text will be organized in the proper reading order.



Microbiol. Immunol., 43(7), 705–709, 1999

Salt-Sensitive Growth of *Staphylococcus aureus*: Stimulation of Salt-Induced Autolysis by Multiple Environmental Factors

Toshiro Ochiai

Faculty of Pharmaceutical Science, Hokuriku University, Kanagawa-machi, Kanazawa, Ishikawa 920–1148, Japan

Received December 3, 1998; in revised form, March 19, 1999. Accepted April 1, 1999

***Abstract*: The growth of *Staphylococcus aureus* 209P became extremely sensitive to a high NaCl concentration following lowered temperature, reduced air-supply, and decreased Ca^{2+} concentration in the medium. Cells in high-NaCl and low-Ca^{2+} concentration media either autolyzed or transformed into protoplast-like forms during growth when grown standing below 37 C. The abnormal growth, however, was invariably avoided by preliminary supplementation with polyanetholesulfonate (autolysin inhibitor) in the growth media. These results suggested that the autolytic activity of this organism was precisely controlled by multiple environmental factors such as ionic strength, temperature, air supply, and the concentration of Ca^{2+}.**

***Key words*: *Staphylococcus aureus*, Salt tolerance, Autolysis, Calcium ion**

Staphylococcus aureus is a salt-tolerant eubacterium, and most strains grow well in a high-NaCl concentration media containing as much as 15% NaCl (1). Certain strains, however, are evidently sensitive to NaCl and exhibit autolysis in the presence of a relatively low concentration of NaCl (4, 10). According to Yabu et al, salt tolerance in salt-sensitive *S. aureus* 209P could easily be restored by autolysin inhibitors such as polyanetholesulfonate (PAS) or lipoteichoic acid (LTA) (10, 11). It therefore seems likely that a salt-sensitive autolytic system is operating in salt-sensitive *S. aureus* (10, 11). Although it has been established that NaCl activates the cell wall-lytic enzymes of staphylococcal cells (2, 8), it is not at all clear why staphylococcal cell wall-lytic enzymes can properly operate under a wide range of ionic strength, and the regulatory system of the lytic enzymes, in itself, still remains equivocal. The present work was undertaken to further investigate the salt-sensitive characteristics of *S. aureus* 209P. The data obtained demonstrated that the autolytic activity of *S. aureus* 209P was affected not only by ionic strength, but also by other environmental factors such as temperature, oxygen supply, and concentration of Ca^{2+}.

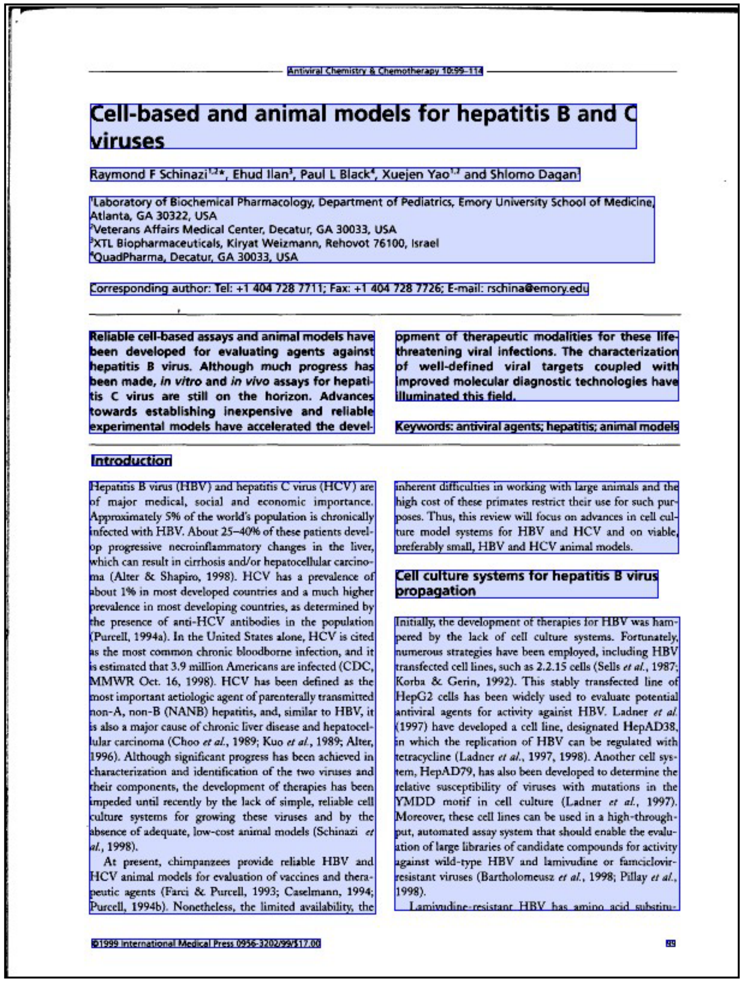
For this study, *S. aureus* 209P (IID671) was used. The stock culture of *S. aureus* 209P was maintained on brain-heart infusion (BHI; Difco Laboratories) agar slopes, stored at 4 C and subcultured monthly. The basal medium used for bacterial cultures was a complex medium (PYKG medium) consisting of 0.5% Bact Peptone (Difco), 0.1% yeast extract (Difco), 0.3% K_2HPO_4, and 0.2% glucose. To reduce the levels of divalent metal ions, PYKG medium was gently stirred at room temperature for 2 hr with the sodium form of Chelex-100 (Bio-Rad Laboratories, Richmond, Calif., U.S.A.). One gram of resin was used per 1 g of dry medium. Following treatment with Chelex-100, the resins were removed by filtration with a glass filter. The levels of Mg^{2+} and Ca^{2+} in the Chelex resin-treated PYKG medium (C-PYKG medium) were 1.1±0.4 and 0.7±0.3 μM, respectively, as determined using an inductively coupled plasma-atomic emission spectrophotometer (Leeman Labs Inc.). The C-PYKG medium was supplemented with NaCl at desired concentrations and the pH was adjusted to 7.2 by adding 10% KOH before autoclaving.

S. aureus 209P did not grow well in C-PYKG medium and the yield obtainable was only about 1/20 of that with PYKG medium (data not shown). However, sufficient growth was easily restored by adding a small amount (final concentration, 50 μM≦) of $MgCl_2$ to the C-PYKG medium. $MgSO_4$ exerted the same effect (data not shown). These observations suggested that the culture still contained small amounts of essential metal ions other than Mg^{2+}. Consequently, C-PYKG medium

Abbreviations: BHI, brain-heart infusion; C-PYKG, Chelex resin-treated PYKG; PAS, polyanetholesulfonate; PYKG, peptone yeast extract supplemented with K_2HPO_4 and glucose; SDS, sodium dodecyl sulfate.

705

Figure 5.4 Correct zones, generated by the zone correction algorithm.



Antiviral Chemistry & Chemotherapy 10:99–114

Cell-based and animal models for hepatitis B and C viruses

Raymond F Schinazi[1,2]*, Ehud Ilan[3], Paul L Black[4], Xuejen Yao[1,2] and Shlomo Dagan[3]

[1]Laboratory of Biochemical Pharmacology, Department of Pediatrics, Emory University School of Medicine, Atlanta, GA 30322, USA
[2]Veterans Affairs Medical Center, Decatur, GA 30033, USA
[3]XTL Biopharmaceuticals, Kiryat Weizmann, Rehovot 76100, Israel
[4]QuadPharma, Decatur, GA 30033, USA

Corresponding author: Tel: +1 404 728 7711; Fax: +1 404 728 7726; E-mail: rschina@emory.edu

Reliable cell-based assays and animal models have been developed for evaluating agents against hepatitis B virus. Although much progress has been made, *in vitro* and *in vivo* assays for hepatitis C virus are still on the horizon. Advances towards establishing inexpensive and reliable experimental models have accelerated the development of therapeutic modalities for these life-threatening viral infections. The characterization of well-defined viral targets coupled with improved molecular diagnostic technologies have illuminated this field.

Keywords: antiviral agents; hepatitis; animal models

Introduction

Hepatitis B virus (HBV) and hepatitis C virus (HCV) are of major medical, social and economic importance. Approximately 5% of the world's population is chronically infected with HBV. About 25–40% of these patients develop progressive necroinflammatory changes in the liver, which can result in cirrhosis and/or hepatocellular carcinoma (Alter & Shapiro, 1998). HCV has a prevalence of about 1% in most developed countries and a much higher prevalence in most developing countries, as determined by the presence of anti-HCV antibodies in the population (Purcell, 1994a). In the United States alone, HCV is cited as the most common chronic bloodborne infection, and it is estimated that 3.9 million Americans are infected (CDC, MMWR Oct. 16, 1998). HCV has been defined as the most important aetiologic agent of parenterally transmitted non-A, non-B (NANB) hepatitis, and, similar to HBV, it is also a major cause of chronic liver disease and hepatocellular carcinoma (Choo *et al.*, 1989; Kuo *et al.*, 1989; Alter, 1996). Although significant progress has been achieved in characterization and identification of the two viruses and their components, the development of therapies has been impeded until recently by the lack of simple, reliable cell culture systems for growing these viruses and by the absence of adequate, low-cost animal models (Schinazi *et al.*, 1998).

At present, chimpanzees provide reliable HBV and HCV animal models for evaluation of vaccines and therapeutic agents (Farci & Purcell, 1993; Caselmann, 1994; Purcell, 1994b). Nonetheless, the limited availability, the inherent difficulties in working with large animals and the high cost of these primates restrict their use for such purposes. Thus, this review will focus on advances in cell culture model systems for HBV and HCV and on viable, preferably small, HBV and HCV animal models.

Cell culture systems for hepatitis B virus propagation

Initially, the development of therapies for HBV was hampered by the lack of cell culture systems. Fortunately, numerous strategies have been employed, including HBV transfected cell lines, such as 2.2.15 cells (Sells *et al.*, 1987; Korba & Gerin, 1992). This stably transfected line of HepG2 cells has been widely used to evaluate potential antiviral agents for activity against HBV. Ladner *et al.* (1997) have developed a cell line, designated HepAD38, in which the replication of HBV can be regulated with tetracycline (Ladner *et al.*, 1997, 1998). Another cell system, HepAD79, has also been developed to determine the relative susceptibility of viruses with mutations in the YMDD motif in cell culture (Ladner *et al.*, 1997). Moreover, these cell lines can be used in a high-throughput, automated assay system that should enable the evaluation of large libraries of candidate compounds for activity against wild-type HBV and lamivudine or famciclovir-resistant viruses (Bartholomeusz *et al.*, 1998; Pillay *et al.*, 1998).

Lamivudine-resistant HBV has amino acid substitu-

©1999 International Medical Press 0956-3202/99/$17.00

99

Figure 5.5 Another example of zones generated by the zone correction algorithm.

5.2 Evaluation of automated zoning

Following initial testing and refinement, the zoning algorithm was tested with a set of page images from 59 journal issues that would become the first set of journals to be processed by the MARS-2 system. Journals selected had a page layout in which the title, authors, affiliations and abstract were all in one column, and appeared on the page in that order. Table 5.2 summarizes the scores for the 295 images in this set. Overall, of the 1,180 possible zones of interest, the zone correction program generated 1,155 correct zones, for a correct rate of 97.9%.

Table 5.2 Results of zone correction for 295 pages from 59 journal issues

Field	Error Type					
	split	too big	too small	merged	totals	% images with an error in this field
Title	7				7	2.4
Author	1			4	5	1.7
Affiliation	4			5	9	3.1
Abstract	3			1	4	1.4
totals	15	0	0	10	25	
% images with this error	5.1	0	0	3.4		

5.3 Implementation

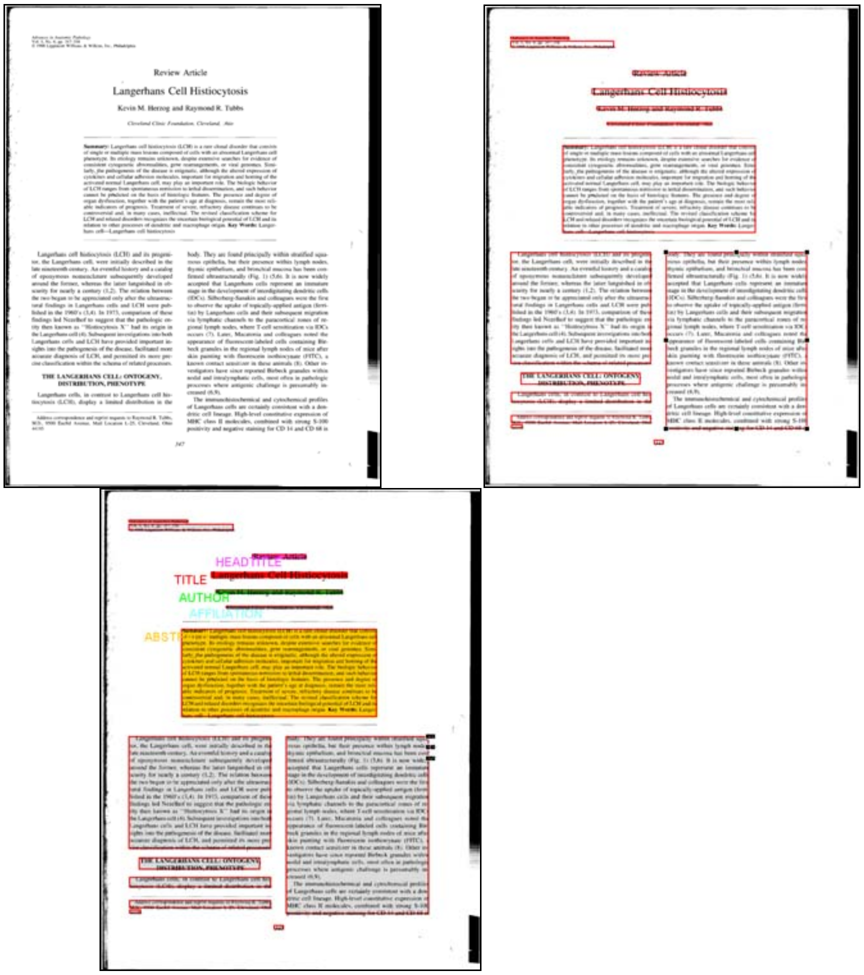
Based on the low error rates achieved in testing, the automatic zone correction algorithm was implemented for the MARS-2 system. A C++ zone correction class was written in the Microsoft Visual Studio development environment. The class is incorporated with the ZoneCzar module that also includes the automated labeling function described in Section 6.

5.4 Performance in production

The original zone correction algorithm has continued to evolve in response to feedback from production operators and to observations from continued internal evaluation. As more journal layout types are added to those processed by MARS-2, code to accommodate new circumstances has been added to the algorithm, but the overall design has not changed. For example, to correctly zone pages in which affiliations are found at or near the bottom of the page, usually in small-sized fonts, computed threshold values are different for lines and zones that begin at the bottom third of the page than they are for the rest of the page. Although performance has remained consistently good, we anticipate challenges as we increase the number of journal titles and layout types accommodated by MARS in the future.

6. Automated Labeling

Once the contiguous text regions in a bitmapped page image are zoned, the next step is to label the zones, i.e., identify each zone as one of the bibliographic fields of interest. The figure below shows the sequence of steps: the bitmapped TIFF image of the scanned page, the output of the automated zoning module (AZ) and the output of the automated labeling module (AL).



(a)

(b)

(c)

Figure 6.1 Results of AZ and AL Modules. (a) Bitmapped page image (b) AZ output, and (c) AL output

Image analysis techniques for document labeling proposed in the literature[33-37] are based mostly on the layout (geometric) structure and/or the logical structure of a document. Hones et al.[33] describe an algorithm for layout extraction of mixed-mode documents, and the classification of these documents as text or non-text. Taylor et al.[34] describe a prototype system using a feature extraction and model-based approach. Tsujimoto et al.[35] present a rule-based technique based on the transformation from a geometric structure to a logical structure. Tateisi et al.[36] propose a method based on stochastic syntactic analysis to extract the logical structure of a printed document. They use simple rules to label documents into three classes. Niyogi et al.[37] use a rule-based system to label newspaper contents into thirteen labels such as headline, text paragraph, photograph, and so on. These labeling techniques rely mostly on rule-based algorithms, but other mechanisms such as artificial neural networks (ANN) and decision trees are also investigated.

One drawback to ANN and decision tree methods is that they need training as a pre-processing stage. That is, the algorithms need to be re-trained whenever a new document (in our case, a journal layout not seen previously) is encountered, and the training time is proportional to the number of journal titles to be processed. Not only is this time consuming, it also makes it difficult for exceptional situations to be handled quickly. In addition, these techniques pose difficulties in readily using geometric information, e.g., the geometry between zones. Rule-based algorithms, on the other hand, do not need re-training, can employ geometric information readily, and moreover, can accommodate exceptional cases (slight divergence from a known layout type) by the addition of new rules. Since the 4,300+ journal titles indexed in MEDLINE exhibit a wide range of layout types, such exceptional cases can occur frequently. An automated labeling system needs to handle a multiplicity of layout types and exceptional cases quickly, and without extensive pre-processing and training.

Our research in this area focused on three approaches: the rule-based algorithmic approach, an ANN method, and a template-matching technique. Our experiments and findings are reported in the literature[38-40]. Based on these experiments, we decided to implement our labeling system on rule-based algorithms since this approach delivered a high accuracy rate, high speed of execution, and furthermore was amenable to modification as new layout types were added.

Our approach relies on data from the OCR system which delivers information at the zone, line and character level:

Zone level Zone boundaries, number of text lines

Line level Line boundaries, number of characters, average character height

Character level 8-bit character code, confidence level (*1= lowest, 9 = highest*), bounding box, font size, font attribute (*normal, bold, underlined, italics, superscript, subscript, and fixed pitch*)

The OCR output data is used to generate geometric and non-geometric features that, in turn, are used to create rules. Geometric features are based on a zone's location, order of appearance, and dimensions. For example, the article title zone is usually located in the top half of the page, followed by author, affiliation and abstract, in that order.

Non-geometric features are derived from the text contents of a zone, aggregate statistics, and font characteristics. For example, some zones can be characterized by the words in them, and the frequency with which they occur. In such cases, word matching is an important technique to generate non-geometric features in the AL module. For example, a zone has a higher probability of being labeled as “affiliation” when it has words representing country, city and school names. Also, a zone positioned between the words “abstract” and “keywords” is more likely to be an abstract than any other bibliographic field. Fifteen database tables containing word lists have been assembled as shown in Table 6.1. Table 6.2 shows examples of geometric and non-geometric features.

Word matching relies on search algorithms such as hash tables, binary search tree, digital search tree, ternary search tree, etc. We chose the ternary search tree on account of its ability to yield both the time efficiency of the digital search tree and the space efficiency of binary search trees, and its ability to perform advanced searches such as partial-matching and near-neighbor search. Proposed by Bentley and Sedgewick in 1997, this technique has been used for several years for searching English dictionaries in a commercial OCR system built at Bell Labs[56].

Table 6.1 Word list tables.

Table Name	Words in the Table
Rubric	Review, Orginal Article, etc.
KeyOfTitle	Study, case, method, etc.
Author	Smith, John, Kim, etc.
AcademicDegree	Ph.D., MD, RN, etc.
Affiliation	University, Department, Institute, etc.
Abstract	Abstract, Summary, Background, etc.
Structured Abstract	Aim, Result, Conclusion, etc.
Keyword	Keyword, Index word, etc.
Received	Received, Revised, Accepted, etc.
Introduction	Introduction, Introduzione, etc.
ExtraDataInAffiliation	Corresponding, Address, To whom, etc.
ExtraDataInLowerAffiliation	Mail, fax, tel, etc.
Date	January, February, 2000, etc.
Publisher	Elsevier, John Wiley, etc.
JournalName	Diabetes, endocrinology, etc.

Table 6.2 Features used in the Automated Labeling module.

Zone Features	Variable Names
Geometric Features:	
Zone coordinates	TopCoordinate, BottomCoordinate, LeftCoordinate, RightCoordinate

Zone height and width	HeightOfZone, LengthOfZone
Median value of height, length and space of lines	MedianLineHeight, MedianLineLength, MedianLineSpace
Difference between the bottom and top coordinates of the bottom-most and top-most zone	HeightOfArticle
Zone order in sequence of top left edge	ZoneOrder
Non-Geometric Features:	
Biggest and smallest font sizes in an article	MaximumFontSize, MinimumFontSize
Number of text lines	NumberOfLine
Number of characters and words	NumberOfCharacter, NumberOfWord
Number of capital characters	NumberOfCapitalCharacter
Dominant font attribute and font size	FontAttribute, FontSize
Confidence of characters	Confidence
Number of "M.D.", "Ph.D.", "RN", etc.	NumberOfDegree
Number of middle names, "Jr", "Sr", "II", etc.	NumberOfMiddleName
Number of city, state, country, school, etc.	NumberOfAffiliation
Number of "abstract", "summary", etc.	NumberOfAbstract
Number of "keywords", "index words", etc.	NumberOfKeyword
Number of "review", "article", etc.	NumberOfHeadtitle
Number of "received", "accepted", etc.	NumberOfReceived
Number of "Introduction", "Introduzione", etc.	NumberOfIntroduction
Percentage of academic degrees per word	PercentOfAcademicDegree
Percentage of middle names per word	PercentOfMiddleName
Percentage of affiliations per word	PercentOfAffiliation
Percentage of capital characters per zone	PercentOfCapitalCharacter

6.1 Definition of layout types

As noted, the MEDLINE database contains bibliographic records from over 4,300 journals. The physical layout of the first page of articles in these journals, and the order in which the five important zones (title, author, upper affiliation, lower affiliation, and abstract) appear on the first page may be used to categorize the zone labeling type for a given journal. Figure 6.2 shows examples of common layout types consisting of a single column, or a combination of single and multiple columns. The numbers in the gray blocks indicate block numbers to help with the definitions of the more common zone labeling types described in Table 6.3.

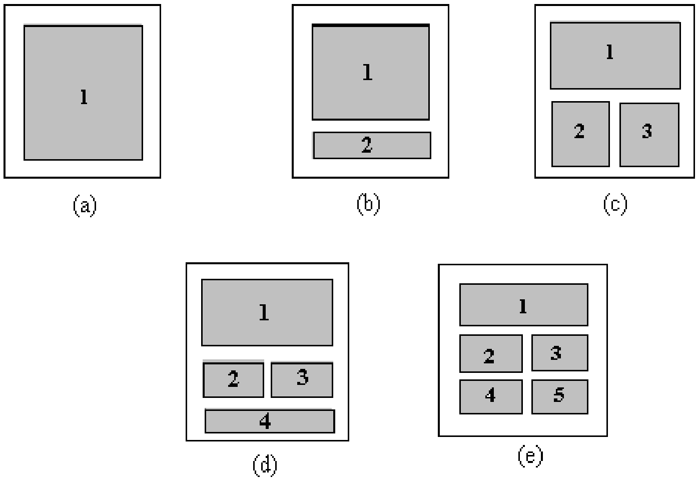


Figure 6.2 Examples of common journal layout types. (a) Layout type 1; (b) Layout type 11; (c) Layout type 12; (d) Layout type 121; (e) Layout type 122.

The five important zones frequently appear in “first regular” or “second regular” zone order. In the “first regular” zone order, the title is near the top of the page, followed by author, affiliation in the upper part of the page (upper affiliation), and abstract. In the “second regular” zone order, the title is followed by author and abstract, with the affiliation appearing in the lower part of the page.

The zone labeling type for each journal is determined by the journal layout type and the zone order. For example, if the journal pages are of layout type 121 [Figure 6.2(d)] and the affiliation appears in block 4 (second regular), the zone labeling type is defined as Type 12006. Other labeling types are described in Table 6.3.

Table 6.3 Description of zone labeling types

Zone Labeling Type	**Includes Layout Type(s)**	**Zone order(s)**	**Description**
Type 10000	1,11,12,121, 122	First regular	Title, author, upper affiliation, and abstract are in block 1.
Type 10006	11	Second regular	Title, author, and abstract are in block 1. Lower affiliation is in block 2.
	121	Second regular	Title, author, and abstract are in block 1. Lower affiliation is in block 4.
Type 12000	12, 121	First regular	Title, author, upper affiliation are in block 1. Abstract is in block 2, and may extend into block 3.
	122	First regular	Title, author, upper affiliation are in block 1. Abstract is in block 2.
Type 12006	121	Second regular	Title and author is in block 1. Lower affiliation is in block 4. Abstract is in block 2, and may extend

			in block 4. Abstract is in block 2, and may extend into block 3.
Type 12200	122	First regular	Title, author, upper affiliation is in block 1. Abstract is in block 2 and 3.

6.2 Structure of AL module

Figure 6.3 shows the structure of the AL module and its interaction with the MARS database whose tables contain information on every journal title (ISSN number). This information includes layout type, physical size, affiliation location, abstract type, feature type, and feature value. After page images from a particular journal issue are processed by the AZ module, and the journal title (ISSN) is identified to the JournalName table, the AL module retrieves all the relevant information from this table, and activates an AL algorithm related to the zone labeling type. The output of the AL module, the identification of the page zones, are written to the LabelRanking table in the database, for further downstream processing.

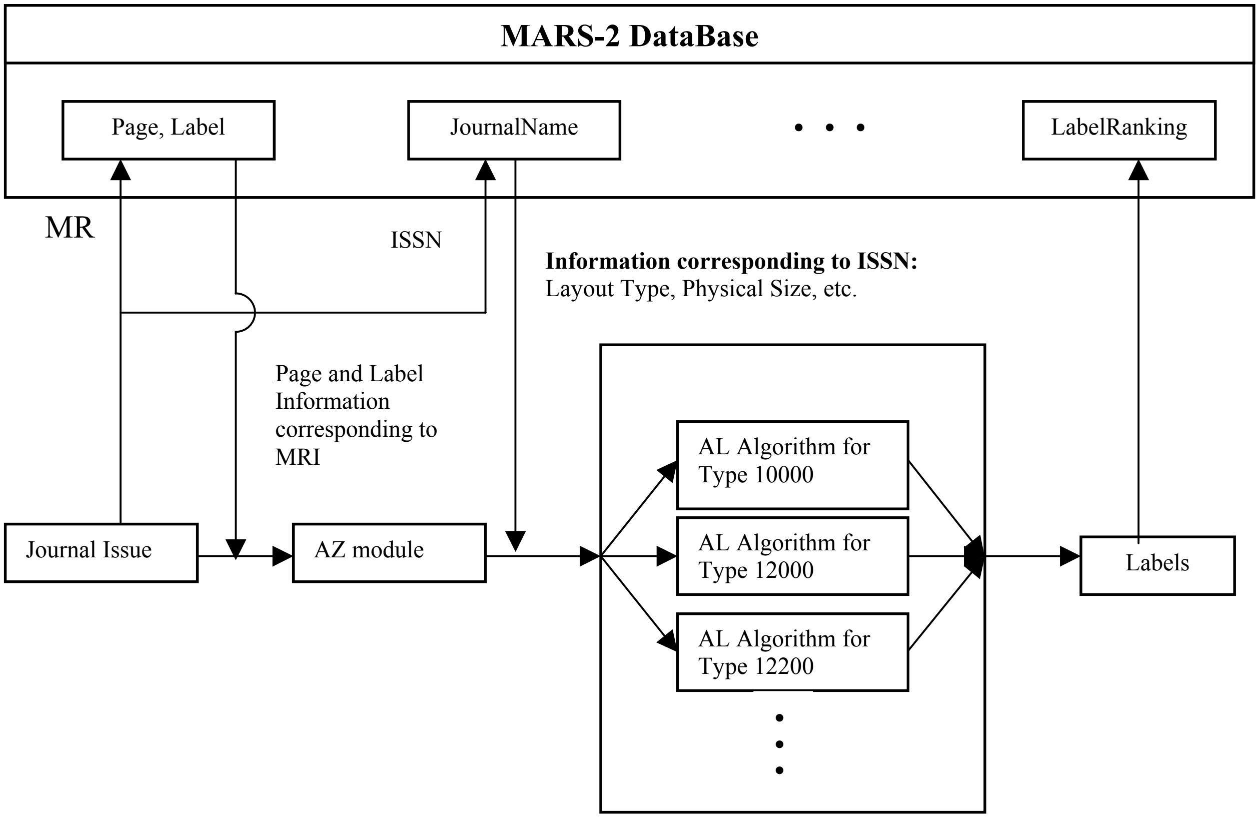

MARS-2 DataBase
Page, Label
JournalName
• • •
LabelRanking
MR
ISSN
Information corresponding to ISSN:
Layout Type, Physical Size, etc.
Page and Label Information corresponding to MRI
Journal Issue
AZ module
AL Algorithm for Type 10000
AL Algorithm for Type 12000
AL Algorithm for Type 12200
Labels

Figure 6.3 Structure of automated labeling module

6.3 Rule-based algorithms in AL module

While all contiguous text regions on a page image are zoned, the only ones of interest in the current MARS system are the article title, author, affiliation and abstract. Since affiliation information could reside in the top part of the page as well as at the bottom, for labeling purposes, we define an “upper affiliation” and a “lower affiliation” zone. Hence, we have five possible labels. The remaining zones are labeled as “other”. For each label type, there are four types of rules as shown in Table 6.3: rule types 1, 2 and 3 that are different for each label classification, and rule type 4 that is the same for all. Our rule-based algorithm consists of four steps.

In the first step, a *probability of correct identification* (PCI) is used in rule type 1. Every zone has five PCIs, one for each label. A PCI is equivalent to the probability of a zone possessing a particular label. The PCIs are derived empirically. For example, in the case of upper affiliation, when more than 30% of words in a zone belong to the affiliation word list, the PCI of upper affiliation is 100. Otherwise, PCI is equal to PercentOfAffiliation ×100/30. In case of author, when more than 28% of words in a zone belong to the list of middle names and academic degrees, the PCI of author is 100. Otherwise, PCI= (PercentOfAcademicDegree + PercentOfMiddleName) × 100/28. In this first step, when a zone has the highest PCI for a particular label, it is assigned that label.

The PCI thresholds of 30% and 28% for affiliation and author respectively are established heuristically. In the case of author, we often find there are two authors in an author zone, each author name usually consists of three words, and "and" is located between the author names. We find that there are middle initials and academic degrees associated with author names. So, a zone is likely to be labeled as author when the ratio of the sum of academic degrees and middle initials to the total number of words in the zone exceeds 2/7 or 28.6%. In the case of affiliations, it has been determined that a zone is likely to be labeled as affiliation when 30% of the words belong to the affiliation word list.

In the second step, the labeling results from step 1 are rechecked by rule type 4. For example, when two zones are both labeled as author but one of those zones is located between title and upper affiliation, and the other is located between upper affiliation and abstract, the latter is removed from the author category.

In the third step, in addition to rule type 2, rule types 1 and 4 are applied again to make sure that at least one zone is labeled as title, author, abstract, upper affiliation or lower affiliation. For example, when a zone initially labeled as author does not contain information relevant to author (NumberOfMiddleName=0 and NumberOfAcademicDegree = 0), its location is then used to do the labeling. That is, its label as author is verified by the facts that (a) it does not contain information related to title or upper affiliation zones, and (b) it is located between title and upper affiliation zones.

In the fourth step, problems caused by zoning errors such as a zone split into multiple zones are handled by all rules, and any remaining unlabeled zones are labeled.

120 rules were generated for zone labeling types 10000, 10006, 12000, 12006, and 12200, and an example of detailed rules to detect upper affiliation is shown Table 6.4.

Table 6.3 Rule Types used in AL Module

Rule Type	Description
1	Use Probability of Correct Identification (PCI). Each label has its own PCI equation. Example: When a zone has a high PCI for a label zone (PCI ≥ 100), the zone is assigned as the label zone.
2	When a label does not have any zone, which has PCI ≥ 100, pick a zone, which has the highest PCI for the label, and assign the zone as the label.
3	Some features should be similar within the same label zones. I.e., when a title zone is divided into two separate zones, the two zones have similar FontSize, FontAttribute, MedianLineHeight, etc.
4	TopCoordinate of title < TopCoordinate of author < TopCoordinate of Upper affiliation < TopCoordinate of abstract author < TopCoordinate of Lower affiliation

Table 6.4 Example of rules to detect Upper Affiliation

Rule Type	Rule Description

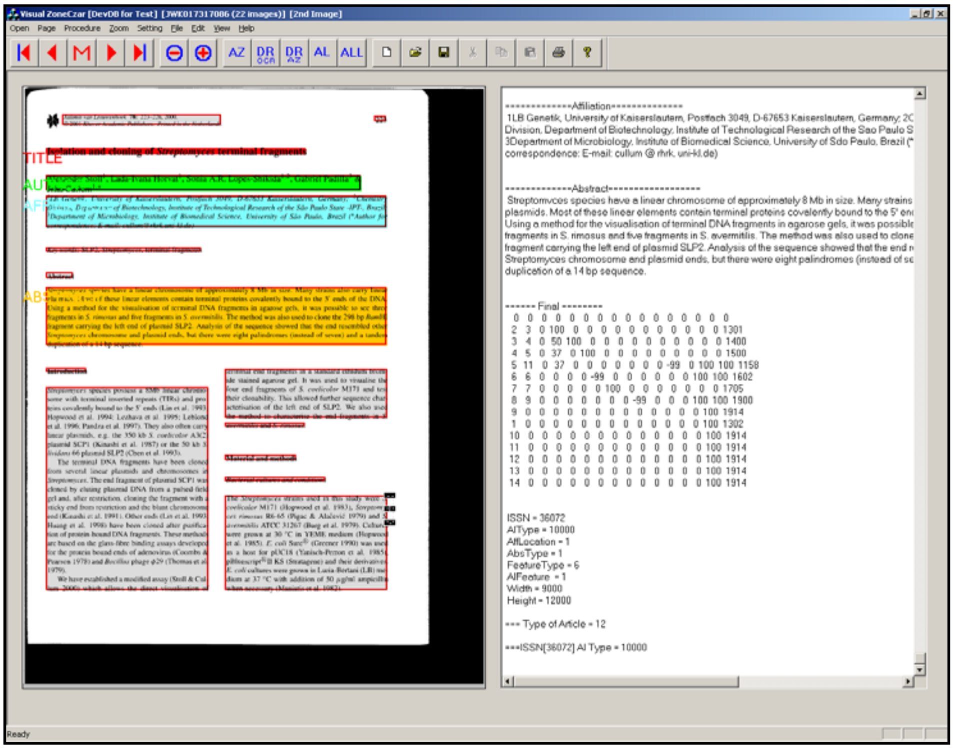
1	1. TopCoordinate < HeightOfArticle /2
	2. BottomCoordinate < HeightOfArticle×3/4
	3. NumberOfWord ≥ 2
	4. NumberOfAcademicDegree < 3 or PercentOfAcademicDegree < 30
	5. NumberOfMiddlename < 3 or PercentOfMiddleName < 30
	6. PercentOfCapitalCharacter < 50
	7. NumberOfHeadtitle == NumberOfAbstract == 0 NumberOfIntroduction==0
	8. If all of above conditions are satisfied {
	    If (NumberOfAffiliation ≥ 2) {
	      If (PercentOfAffiliation ≥ 30) PCI =100;
	      Else PCI = PercentOfAffiliation×100/30;
	    }
	    Else {
	      If (PercentOfAffiliation ≥ 30) PCI =50;
	      Else PCI = PercentOfAffiliation×50/30;
	    }
	  }
	  Else {
	    PCI = 0
	  }

2	If (PCI < 100), pick a zone having the highest PCI for upper affiliation.
3	1.If (PCI > 25 and the next zone has NumberOfReceived ==1) PCI = 100.

	2.Distance from a zone to upper affiliation zone is smaller than any other label zones. 3.FontSize, FontAttribute, MedianOfLineHeight, and MedianOfLineSpace of a zone must be similar to upper affiliation zone.
4	TopCoordinate of title < TopCoordinate of author < TopCoordinate of affiliation < TopCoordinate of abstract

6.4 Research tool for labeling

As noted earlier, the zoning and labeling functions are integrated into one module, ZoneCzar. Since this is a daemon, there is no operator workstation. However, a GUI is provided for a supervisor or the development team to check on problems or progress. Apart from this, we created a research tool, Visual ZoneCzar, to help develop and test the algorithms used for labeling. Its design is based on Visual C++(6.0).


Visual ZoneCzar [DevDB for Test] [JWK017317006 (22 images)] [2nd Image]
Open Page Procedure Zoom Setting File Edit View Help
M
AZ
DR OCR
DR AZ
AL
ALL
TITLE
AU
AF
AB
Ready

```
--------------Affiliation---------------
 1LB Genetik, University of Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern, Germany; 2C
Division, Department of Biotechnology, Institute of Technological Research of the Sao Paulo S
3Department of Microbiology, Institute of Biomedical Science, University of Sdo Paulo, Brazil (*
correspondence: E-mail: cullum @ rhrk. uni-kl.de)

--------------Abstract-------------------
 Streptomvces species have a linear chromosome of approximately 8 Mb in size. Many strains
plasmids. Most of these linear elements contain terminal proteins covalently bound to the 5' en
Using a method for the visualisation of terminal DNA fragments in agarose gels, it was possibl
fragments in S. rimosus and five fragments in S. avermitilis. The method was also used to clone
fragment carrying the left end of plasmid SLP2. Analysis of the sequence showed that the end r
Streptomyces chromosome and plasmid ends, but there were eight palindromes (instead of sc
duplication of a 14 bp sequence.

------- Final ---------
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  2   3   0 100   0   0   0   0   0   0   0   0   0   0   0 1301
  3   4   0  50 100   0   0   0   0   0   0   0   0   0   0 1400
  4   5   0  37   0 100   0   0   0   0   0   0   0   0   0 1500
  5  11   0  37   0   0   0   0   0   0   0 -99   0 100 100 1158
  6   6   0   0   0   0 -99   0   0   0   0   0   0 100 100 1602
  7   7   0   0   0   0   0 100   0   0   0   0   0   0   0 1705
  8   9   0   0   0   0   0   0   0 -99   0   0   0 100 100 1900
  9   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
  1   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1302
 10   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 11   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 12   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 13   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914
 14   0   0   0   0   0   0   0   0   0   0   0   0   0 100 1914

 ISSN = 36072
 AlType = 10000
 AffLocation = 1
 AbsType = 1
 FeatureType = 6
 AlFeature  = 1
 Width = 9000
 Height = 12000

=== Type of Article = 12

===ISSN[36072] Al Type = 10000
```

Figure 6.4 GUI of Visual ZoneCzar

The purpose of Visual ZoneCzar is to test the algorithms on page images from a new journal title to be included in the list of journal titles that may be processed automatically by MARS-2. If the existing algorithms fail to successfully label the zones from the new journal, rules are modified, and tests are repeated. This tool helps the researcher check and verify that the algorithmic rules are in fact applicable to a particular journal.

As shown in Figure 6.4, the GUI of Visual ZoneCzar has two windows. The left window displays zoning and labeling results on a TIFF image. The zones are displayed by red colored boundaries. The labeling results are displayed by different background colors and text. For example, the article title zone is red accompanied by the word TITLE in red, and the author zone is green with the word AUTHOR in green, and so on. Zones that are not of interest are in gray.

The right window displays the text in the zones that are shown labeled in the left window, and the labeling rules in the algorithm. In the example, the text in the zones that have been labeled as affiliation and abstract are shown. The 14x17 table in the middle of the window gives information on the rules being applied. A summary follows describing the contents of this table.

The row number in the table corresponds to the ordinal number of zone number in the OCR output data. The first column indicates zone number; the second column contains a number representing labeling results. In the second column, for example, the “3” means that zone 2 is labeled as a title. Other numbers (4, 5 or 7) would refer to author, upper affiliation, and abstract labels. The third to the thirteenth column shows the calculated PCIs for rubric, title, author, upper affiliation, word abstract, abstract, keyword, introduction, lower affiliation, upper received, and lower received for each zone. A PCI of 100 means that the zone is labeled as one of the five important labels which are title, author, upper affiliation, lower affiliation, or abstract. For the other labels (not important to the MARS system), a PCI of –99 is assigned. The fourteenth column has "100" when the zone was assigned a PCI = -99. The fifteenth column has "100" when the zone is none of the identifiable zones. The sixteenth column shows the rule number used to label the zone. In the case of the second row, there are "2", "3", "100", and "1301" in the first, second, fourth, and sixteenth columns. This is shorthand indicating that the zone number two is labeled as title by rule 1301. The third row shows that zone number three is labeled as author by rule 1400, and has PCI =50 to title label. The eighth row shows that zone number eight is labeled as introduction by rule 1900.

A four-digit number is used to identify a rule. The highest digit indicates the step of labeling process, the next digit indicates the label, and the lowest two digits indicate the rule number. For example, 1301 in the second row means that in step one (1) of the labeling procedure for the title label (3), rule one (01) was used.

Other information about the journal issue (MRI) obtained from the JournalName table in the database is displayed at the bottom of the right window.

The tool bar of Visual ZoneCzar offers the researcher twelve buttons to navigate and control the data. The first button displays the first page image in the journal issue, the second button displays the previous page, the third button displays a page in the middle of the group of pages, the fourth button displays the next page, and the fifth button displays the last page. The sixth and seventh buttons are to minimize and maximize the TIFF images. The eighth through the twelfth buttons control the zoning and labeling process. The AZ button runs the zoning module, DR OCR and DR AZ buttons display the OCR and zoning results, AL runs the labeling process, and ALL runs both zoning and labeling modules.

6.5 Performance in production

Currently the AL module can reliably process 2,027 journal titles from the 4,300+ titles indexed in MEDLINE. Since NLM receives bibliographic data for 580 of these directly from publishers, the actual number of titles that may be processed by MARS-2 is 1,447.

In Table 6.5 we show performance data for the month of February 2001 for 159 journal issues containing 2,524 articles processed by MARS-2. This collection exhibited four layout types. There were 101, 10, 37 and 11 journal issues in zone labeling types 10000, 10006, 12000, and 12200 respectively.

The data shows that 0.4% of the labeling errors is due to incorrect OCR output and 0.63% is due to poor zoning (AZ). The error rate attributed to the AL module itself is 0.20% when OCR and AZ are correct. The reason for the high error rate in the affiliation field is that text in this field is small sized and are frequently italicized, both factors contributing to poor detection by the OCR system. In overall performance, the AL module delivers an accuracy of 98.77%.

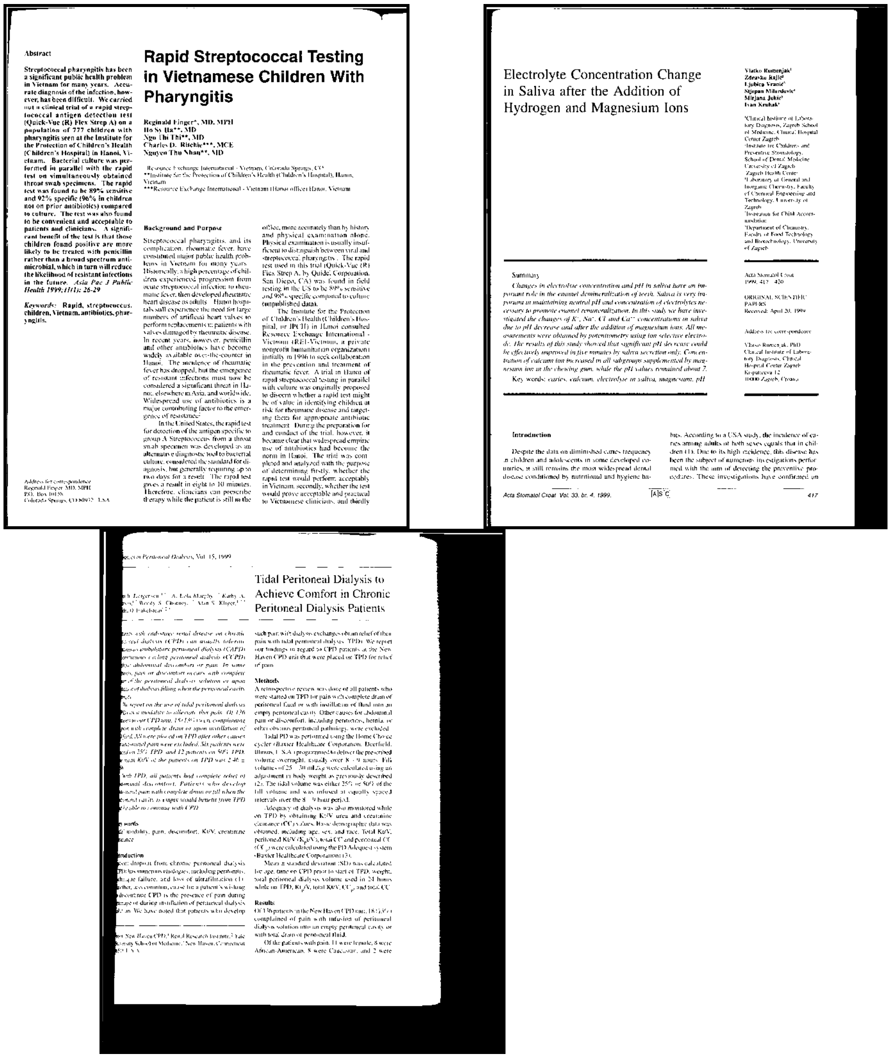
Table 6.5 Automatic labeling performance

Error Type	**Label Field**					
	Title	**Author**	**Affiliation**	**Abstract**	**Totals**	**% of Error**
Bad OCR	0	1	9	0	10	0.40
Automated Zoning (AZ)	2	8	6	0	16	0.63
Automated Labeling (AL)	1	3	1	0	5	0.20
Totals	3	12	16	0	31	1.23
% of Error	0.12	0.48	0.63	0	1.23	

6.6 Ongoing research

As mentioned earlier, we used empirical methods to derive thresholds for the probability of correct identification (PCI) for each label, such as 28% and 30% of special word lists for PCI thresholds for author and affiliation. We plan to refine these figures by using statistical data, i.e., create histograms of every word list collected from the journals processed by MARS-2 for each label zone, and select thresholds based on these histograms.

We are continually increasing the number of journal titles accommodated by MARS-2, but we find that a number of these do not follow the relatively regular layout types that the system can process at present. Figure 6.5 shows examples of these irregular layouts. Figure 6.5(a) has the abstract to the left of the article title, Figure 6.5(b) has author and affiliation to the right of the title, and Figure 6.5(c) has author to the left of the title, all quite different from the "regular" layouts. One approach to dealing with these irregular layouts is to develop a template matching algorithm based on the average font size and the average top-left and bottom-right coordinates of all important zones. These features will be stored in the database in a journal-specific manner. When a journal issue with irregular layout is processed, the AL module will read the zone coordinates and the font size of the text in the zone, and match them against the stored information.



Abstract

Streptococcal pharyngitis has been a significant public health problem in Vietnam for many years. Accurate diagnosis of the infection, however, has been difficult. We carried out a clinical trial of a rapid streptococcal antigen detection test (Quick-Vue (R) Flex Strep A) on a population of 777 children with pharyngitis seen at the Institute for the Protection of Children's Health (Children's Hospital) in Hanoi, Vietnam. Bacterial culture was performed in parallel with the rapid test on simultaneously obtained throat swab specimens. The rapid test was found to be 89% sensitive and 92% specific (96% in children not on prior antibiotics) compared to culture. The test was also found to be convenient and acceptable to patients and clinicians. A significant benefit of the test is that those children found positive are more likely to be treated with penicillin rather than a broad spectrum antimicrobial, which in turn will reduce the likelihood of resistant infections in the future. *Asia Pac J Public Health 1999;11(1): 26-29*

Keywords: Rapid, streptococcus, children, Vietnam, antibiotics, pharyngitis.

Address for correspondence
Reginald Finger MD, MPH
P.O. Box 10158
Colorado Springs, CO 80932 USA

Rapid Streptococcal Testing in Vietnamese Children With Pharyngitis

Reginald Finger*, MD, MPH
Ho Sy Ha**, MD
Ngo Thi Thi**, MD
Charles D. Ritchie***, MCE
Nguyen Thu Nhan**, MD

*Resource Exchange International - Vietnam, Colorado Springs, CO
**Institute for the Protection of Children's Health (Children's Hospital), Hanoi, Vietnam
***Resource Exchange International - Vietnam (Hanoi office) Hanoi, Vietnam

Background and Purpose

Streptococcal pharyngitis, and its complication, rheumatic fever, have constituted major public health problems in Vietnam for many years. Historically, a high percentage of children experienced progression from acute streptococcal infection to rheumatic fever, then developed rheumatic heart disease as adults. Hanoi hospitals still experience the need for large numbers of artificial heart valves to perform replacements in patients with valves damaged by rheumatic disease. In recent years, however, penicillin and other antibiotics have become widely available over-the-counter in Hanoi. The incidence of rheumatic fever has dropped, but the emergence of resistant infections must now be considered a significant threat in Hanoi, elsewhere in Asia, and worldwide. Widespread use of antibiotics is a major contributing factor to the emergence of resistance.

In the United States, the rapid test for detection of the antigen specific to group A Streptococcus from a throat swab specimen was developed as an alternative diagnostic tool to bacterial culture, considered the standard for diagnosis, but generally requiring up to two days for a result. The rapid test gives a result in eight to 10 minutes. Therefore, clinicians can prescribe therapy while the patient is still in the office, more accurately than by history and physical examination alone. Physical examination is usually insufficient to distinguish between viral and streptococcal pharyngitis. The rapid test used in this trial (Quick-Vue (R) Flex Strep A, by Quidel Corporation, San Diego, CA) was found in field testing in the US to be 89% sensitive and 98% specific compared to culture (unpublished data).

The Institute for the Protection of Children's Health (Children's Hospital, or IPCH) in Hanoi consulted Resource Exchange International - Vietnam (REI-Vietnam), a private nonprofit humanitarian organization) initially in 1996 to seek collaboration in the prevention and treatment of rheumatic fever. A trial in Hanoi of rapid streptococcal testing in parallel with culture was originally proposed to discern whether a rapid test might be of value in identifying children at risk for rheumatic disease and targeting them for appropriate antibiotic treatment. During the preparation for and conduct of the trial, however, it became clear that widespread empiric use of antibiotics had become the norm in Hanoi. The trial was completed and analyzed with the purpose of determining firstly, whether the rapid test would perform acceptably in Vietnam, secondly, whether the test would prove acceptable and practical to Vietnamese clinicians, and thirdly

Electrolyte Concentration Change in Saliva after the Addition of Hydrogen and Magnesium Ions

Vlatko Rumenjak[1]
Zdravko Rajić[2]
Ljubica Vranić[3]
Stjepan Milardović[4]
Mirjana Jukić[5]
Ivan Kruhak[1]

[1]Clinical Institute of Laboratory Diagnosis, Zagreb School of Medicine, Clinical Hospital Center Zagreb
[2]Institute for Children's and Preventive Stomatology, School of Dental Medicine University of Zagreb
[3]Zagreb Health Center
[4]Laboratory of General and Inorganic Chemistry, Faculty of Chemical Engineering and Technology, University of Zagreb
[5]Institution for Child Accommodation
[6]Department of Chemistry, Faculty of Food Technology and Biotechnology, University of Zagreb

Summary

Changes in electrolyte concentration and pH in saliva have an important role in the enamel demineralization of teeth. Saliva is very important in maintaining neutral pH and concentration of electrolytes necessary to promote enamel remineralization. In this study we have investigated the changes of K^+, Na^+, Cl^- and Ca^{++} concentrations in saliva due to pH decrease and after the addition of magnesium ions. All measurements were obtained by potentiometry using ion selective electrode. The results of this study showed that significant pH decrease could be effectively improved in five minutes by saliva secretion only. Concentration of calcium ion increased in all subgroups supplemented by magnesium ion in the chewing gum, while the pH values remained about 7.

Key words: *caries, calcium, electrolyte in saliva, magnesium, pH*

Acta Stomatol Croat
1999; 417 - 420

ORIGINAL SCIENTIFIC PAPERS
Received: April 20, 1999

Address for correspondence

Vlatko Rumenjak, PhD
Clinical Institute of Laboratory Diagnosis, Clinical Hospital Center Zagreb
Kispatićeva 12
10000 Zagreb, Croatia

Introduction

Despite the data on diminished caries frequency in children and adolescents in some developed countries, it still remains the most widespread dental disease conditioned by nutritional and hygiene habits. According to a USA study, the incidence of caries among adults of both sexes equals that in children (1). Due to its high incidence, this disease has been the subject of numerous investigations performed with the aim of detecting the preventive procedures. These investigations have confirmed an

Acta Stomatol Croat Vol. 33, br. 4, 1999. [A]S C 417

ces in Peritoneal Dialysis, Vol. 15, 1999

Tidal Peritoneal Dialysis to Achieve Comfort in Chronic Peritoneal Dialysis Patients

h Jorgensen,[1,2] A. Lola Murphy,[1] Kathy A. Leo,[1] Wendy S. Chorney,[1] Alan S. Kliger,[1,2] d O. Finkelstein[2,3]

nts with end-stage renal disease on chronic al dialysis (CPD) can usually tolerate ambulatory peritoneal dialysis (CAPD) continuous cyclic peritoneal dialysis (CCPD) abdominal discomfort or pain. In some pain or discomfort occurs with complete of the peritoneal dialysis solution or upon of dialysis filling when the peritoneal cavity

We report on the use of tidal peritoneal dialysis as a modality to alleviate this pain. Of 136 in our CPD unit, 18 (13%) were complaining with complete drain or upon instillation of fluid. All were placed on TPD after other causes abdominal pain were excluded. Six patients were on 25% TPD and 12 patients on 50% TPD. mean Kt/V of the patients on TPD was 2.46 ±

with TPD, all patients had complete relief of abdominal discomfort. Patients who develop abdominal pain with complete drain or fill when the peritoneal cavity is empty would benefit from TPD be able to continue with CPD.

Key words
Tidal modality, pain, discomfort, Kt/V, creatinine clearance

Introduction
Patient dropout from chronic peritoneal dialysis (CPD) has numerous etiologies, including peritonitis, technique failure, and loss of ultrafiltration (1). Another, less common, cause for a patient's wishing to discontinue CPD is the presence of pain during drainage or during instillation of peritoneal dialysis (2). We have noted that patients who develop such pain with dialysis exchanges obtain relief of their pain with tidal peritoneal dialysis (TPD). We report our findings in regard to CPD patients at the New Haven CPD unit that were placed on TPD for relief of pain.

Methods
A retrospective review was done of all patients who were started on TPD for pain with complete drain of peritoneal fluid or with instillation of fluid into an empty peritoneal cavity. Other causes for abdominal pain or discomfort, including peritonitis, hernia, or other obvious peritoneal pathology, were excluded.

Tidal PD was performed using the Home Choice cycler (Baxter Healthcare Corporation, Deerfield, Illinois, U.S.A.) programmed to deliver the prescribed volume overnight, usually over 8 - 9 hours. Fill volumes of 25 - 30 mL/kg were calculated using an adjustment in body weight as previously described (2). The tidal volume was either 25% or 50% of the fill volume and was infused at equally spaced intervals over the 8 - 9 hour period.

Adequacy of dialysis was also monitored while on TPD by obtaining Kt/V urea and creatinine clearance (CC) values. Basic demographic data was obtained, including age, sex, and race. Total Kt/V, peritoneal Kt/V (K_pt/V), total CC and peritoneal CC (CC_p) were calculated using the PD Adequest system (Baxter Healthcare Corporation) (3).

Mean ± standard deviation (SD) was calculated for age, time on CPD prior to start of TPD, weight, total peritoneal dialysis volume used in 24 hours while on TPD, Kt_p/V, total Kt/V, CC_p, and total CC.

Results
Of 136 patients in the New Haven CPD unit, 18 (13.3%) complained of pain with infusion of peritoneal dialysis solution into an empty peritoneal cavity or with total drain of peritoneal fluid.

Of the patients with pain, 11 were female, 8 were African-American, 8 were Caucasian, and 2 were

From: [1]New Haven CPD,[1] Renal Research Institute,[2] Yale University School of Medicine,[3] New Haven, Connecticut, U.S.A.

(a)

(b)

(c)

Figure 6.5 Examples of journals exhibiting irregular layout. (a) Abstract is located to the left of the title. (b) Author and affiliation are located to the right of the title. (c) Author is located to the left of the title.

7 Automated reformatting

Following the labeling of the zoned text, the contents of each zone corresponds to the article title, author names, affiliation, and abstract that we are seeking. However, the text in the first three zones is rarely in the syntactic forms required by MEDLINE's conventions. The automated reformatting stage (Autoreformat) is designed to convert this text to the desired formats to eliminate manual correction at the reconcile stage.

The reformatting of the title and author fields is implemented by predefined rules. Rules for the title field implement retaining the capital case for the first letter of the first word, and the de-capitalization of all the other words *with the exception of acronyms*. An example: "Medical Management of AIDS Patients" becomes "Medical management of AIDS patients," as required in MEDLINE. Rules for author fields take into account characters that delimit authors in a multiple-author list; tokens to be eliminated, such as Ph.D., M.D.; tokens to be converted, such as II to 2nd; and "particles" to be retained, such as "van." For example, the author name appearing on the printed page as *Eric S. van Bueron, Ph.D.* becomes *Van Bueron ES* as required in MEDLINE.

Based on journal title and label (author or title), the reformatting module selects a subset of rules from the inclusive set of all rules. The selected rule set and the OCR output text are passed to the reformatting algorithm, and as each rule is applied, the OCR string is modified. Our experiments before implementing this function in the production system correctly reformatted more than 97% of the authors and titles from a test set of 1,857 processed articles. This performance may be expected to improve with the addition of rules derived from production data.

The reformatting strategy for the affiliation field is quite different from the above. The OCR data for an affiliation field could contain many affiliations, since each author may have a different affiliation. This data is often difficult to reformat. One reason is that only the affiliation of the first author is to be retained, in line with MEDLINE conventions. Another reason is that the desired data is spread out over the entire field and not contiguous. For example, in a 30 word affiliation zone, we may only want to retain words 1-8, 12-14, and word 30. Our method involves probability matching of the OCR output text to historical data of ~130,000 unique affiliations.

In the case of affiliations, in addition to the processing at the reformat stage, we attempt to improve the recognition of affiliations by lexicon-based methods described in Section 8.

7.1 Reformatting the Author field

Reformatting the author field uses forward chaining rules-based deduction. The reformat module can have many rules defined for a particular field. Each rule has a number of requirements among which are that it must

- Be associated with a specific journal title (ISSN number);

- Fall into one of eight categories as listed in Table 7.1. The categories are pre-defined in the reformat module and are required to help in our conflict resolution strategy, which in our case is *specificity ordering.* Whenever the conditions of one triggering rule is a superset of another rule, the superset rule takes precedence in that it deals with more specific situations. An example of this is shown later.

The example column in Table 7.1 shows the complete reformatted field. Note that a single rule or category does not necessarily complete the reformatting, but may need to be combined to achieve correct reformatting of the author field.

With the eight categories defined, the first step in using the reformat module for a given ISSN is to define which rules are appropriate for a particular ISSN (journal title), since the printed format varies widely among journals. As an example, in one journal the authors appear as:

Glenn M Ford, MD, John Smith, PhD, and John Glover

This can be difficult to parse with a default set of rules, such as ', and' and ',' so that other rules need to be defined. By defining, in the database, the rules for a specific journal title over a specific period[1] of time we can customize the rules to work for unusual or specific cases.

The above example fails in the default rule set that only has ',' and ', and' as the author delimiters because this would incorrectly identify 'MD' and 'PhD' as author names. To accommodate this journal (and others like it) a high priority rule trigger list was created for author delimiters such as ', MD', ', PhD', 'Mr.', 'Dr.', and other formal titles.

To avoid conflict among rules, each word chain is passed through all the categories recursively until no more rules are triggered. As long as we have an antecedent with consequences we continue to process the word chain. Using the forwarding chaining method, when an "if statement" is observed to match an assertion, the antecedent (i.e., the if statement) is satisfied. When the entire set of if statements are satisfied, the rule is triggered. Each rule that is triggered establishes, in a working memory node, that it was executed. During conflict resolution the reformat module decides which rules take priority over others via specificity ordering. An example would be:

Reduce category executes on 'John Smith II' and makes this 'J S II'
Convert category executes 'John Smith II' and marks Smith as convert pre-word and 'II' to '2nd'.

Our conflict resolution method specifies that the convert category is more specific than the reduce category, thus keeping the word 'Smith' and '2nd'. In addition, the pre-word convert flag in this particular example signals the conflict resolution manager to keep 'Smith', initialize 'J', and append '2nd '. This is possible because we have retained our

[1] Journals often change formats over the years to accommodate new publishers or printers. Therefore the rules may need to change even though the journal title remains the same.

original text and the converted text. The text did not change and an integrated rule has informed us that the word 'Smith' has remained unchanged, and by examining all words, we deduce that this is the last name.

Example Before/After:
Before - John Smith II
After - Smith J 2nd

At the category level, the conflict resolution strategy is specificity ordering. There is also a conflict resolution strategy within a given category: priority list rule ordering. Rules within a given category are assigned a priority level to avoid conflicts. An example of this is the following list of authors appearing on the printed page:

Glenn Ford, John Smith, and David Wells

We have the following author delimiter rules defined

',' and ', and'

However, the ',' is assigned priority 1, and the ', and' is assigned a higher priority 2. If we did not give a higher priority to ', and' we could end up with 'and' as part of the author name or create a null value.

In ground truth testing of the author reformat rules system we tested 1,857 authors from OCR data. Of that number, 41 were reformatted incorrectly, for a 97.29% correction rate. Of those 41, all 41 were missing rules defined for a given case. An example of a missing rule is given in the case of an author field that reads:

Glenn M. Ford, Jr., John Smith.

By simply adding the rule [', Jr. ' author delimiter priority 2], and with no changes in code, we achieved 100% correct reformatting in the test set.

7.2 Reformatting the Article Title field

The title field uses the same principles as in the author rules system, but requires fewer rules or categories. Of the eight rule categories required for reformatting authors, only three are needed to reformat titles: Uppercase, Lowercase and First Letter Upper.

7.3 Reformatting the Affiliation field

Institutional affiliations of the authors are reformatted by finding the best match between the OCR text and a list of about 130,000 correctly formatted affiliations obtained from the current production version of MARS. Simple string matching is not promising because of the myriad arrangements in which affiliations can be expressed. Most journals show the affiliations of all authors, but by convention only the affiliation of the first author is entered into MEDLINE. However, the text string corresponding to the first affiliation may be scattered throughout the OCR text for the affiliation field. As an example, when multiple authors are affiliated with different departments within the same institution, the printed affiliation may be "Department A, Department B, Department C, Institution XYZ," while the correct MEDLINE entry is "Department A, Institution XYZ." The problem is further confounded by OCR errors, especially errors in detecting superscripts and subscripts. To find a match, the entire OCR text of the affiliation field is compared with every entry in the list of existing affiliations. A matching score for each of the existing affiliations is calculated on the basis of partial token matches, distance between token matches and customized soundex matching. The three highest scoring candidates are presented to the Reconcile operator for selection. In preliminary tests, our current version of affiliation field reformatting successfully identifies the correct affiliation over 80% of the time when the affiliation is represented in the list. This success rate is expected to improve with parallel efforts to reduce OCR errors and the expansion of the list of affiliations from ongoing production data.

The first step is to read all these unique affiliations into memory and create a ternary search tree[56] for each affiliation, after which we create a soundex word list[57] for each affiliation.

When a zone is identified at the labeling stage as an affiliation field, the OCR data is first processed through a partial-matching algorithm. Low confidence characters are replaced with wildcards.

> Example: Uni**u**ersity. The 'u' is actually a 'v' but the OCR engine assigned it as a 'u' with a low confidence level. The partial match algorithm replaces the 'u' with a '.' signifying that this character is a wildcard, and that any word in our search tree that has the pattern Uni<any letter>ersity is considered to be a match.

The first step is to determine if a word in the affiliation zone matches one in the affiliation list. Ignoring implemented performance optimizations[2] we perform a partial word match for all the words in the OCR list and build up a chain of those words that do match. We also track distances between chains.

Consider the example of trying to find the affiliation "Department of Computer Science, University of Maryland" in the affiliation list. The OCR input string might look like: "Department of Computer Science, Department of Engineering, University of Maryland, Department of Computer Science, Johns Hopkins University."

Since only the first affiliation is to be retained, there is considerable data that is irrelevant. The problem is to retrieve just the data needed. By word chaining we can find chains of words that exist in both the OCR text and in an affiliation zone and then use these to derive weighted probabilities.

[2] Optimizations such as: if the first word does not exist in the affiliation listing entry 1, go to entry 2 instead of looking at every OCR word.

In this example there is a chain of 4 words that match, followed by 3 that do not match, followed by 3 more that match, and finally 7 that do not. Our probability algorithms compute chain word matches and distances between chained words.

The next step in our process reverses the partial word match. The ~130,000 affiliations are matched to the OCR affiliation.

Using the same example, "Department of Computer Science, University of Maryland" has 7 words and all 7 occur in our OCR word list. It is likely there is another affiliation entry that looks like "Department of Computer Science, University of Delaware". This would give a high match of 6/7 words. By comparing and weighting word matches from OCR to Corrected Affiliation and Corrected Affiliation to OCR, and using information such as the number of words matched, total number of words, chain of words matched, and chain of words unmatched, we arrive at a probability between 0 and 1. Note that partial matching is used to help cover OCR errors that would ruin a literal string pattern matching as the affiliation field is often in a smaller font and is likely to incur higher than normal OCR error rates.

In addition to a partial match search algorithm, a soundex algorithm is used with the addition of OCR substitution. For the example in which 'Uni**u**ersity" has the 'u' as low confidence, a substitution table developed lists of common OCR errors where a u == v == y. All three letters are substituted in the low confidence 'u' position, and if a word matches with a soundex hash it counts as a match.

In our ground truth testing with affiliation zones[23], we found that if the OCR affiliation exists in our affiliation list of 130,000 entries, the probability that the affiliation match is the correct one is 88%. The affiliation reformat module picks the top 5 candidates which are presented to the reconcile operator who can choose the correct one in the 5, or pick the nearest match and type in any missing data, usually a room number, zip code or an email address.

7.4 Ongoing work

Current research focuses on the correct detection of superscripts in both the author and affiliation fields to help improve reformatting algorithms. With this information available, correct affiliation matching is expected to improve further.

Table 7.1 Categories of Author Reformat Rules

Category	**Description**	**Example**
Particle Name	Many names contain "particles" forming an integral part of the family name and possibly bearing significance to the family. A particle is retained as part of the reformatted author name.	*Etienne du Vivier* becomes *du Vivier E,* where 'du' is a particle and is retained as is and preceding the last name Vivier. The first name is initialized.
Compound	Compound family names are preserved in the form given and are often difficult to detect. We use a mix of rules to deduce it as a compound name. Most compound names use a hyphen. Those that don't can often use particle name rules to help preserve the compound name.	*L.G. Huis in 't Veld* becomes *Huis in 't Veld LG* *H.G. Huigbregtse-Meyerink* becomes *HuigBregtse-Meyerink HG*
Convert	Convert is a broad category that deals with general requirements to convert one pattern of text with another.	James A. Smith IV becomes Smith JA 4th
Religious	Religious titles include Mother, Sister, Father, Brother. Names with surnames are handled differently from those that have no surnames.	Surname example: *Sister Mary Hilda Miley* becomes *Miley MH* No-Surname example: *Sister May Hilda* becomes *Mary Hilda Sister* For translated articles, e.g., from the French, *Soeur* becomes *Sister*.
Reduce	Reduction rules cover the elimination of text with a single author name. It also handles the Reduction of a person's given name and marking of the Surname if present.	*Mr. John Smith* becomes *Smith J* *John Smith MD* becomes **Smith J**
Lowercase	Some fields present all data uppercase. This rule simply converts to lower case all text that is uppercase.	JOHN SMITH becomes *Smith J*
First Letter	Title and Author at times will require that	JOHN SMITH becomes

Upper	the first letter of a specific word be uppercased, depending on other rules.	*Smith J*
Author Delimiter	Many articles are by multiple authors who contributed to the paper, such as this one. This rule takes an OCR stream of text and creates a word list, a chain of words, and delimits where a particular author begins and ends in the complete chain of words.	Example1: **Glenn M Ford, John Smith** becomes: *Ford GM* *Smith J* (, is the delimiter here) Example 2: *Glenn M. Ford, John Smith, and Susan O'Malley* becomes: *Ford GM* *Smith J* *O’Malley S* (', and' is the trigger, which must precede in priority ',' as a triggered rule)

8 Lexical analysis to improve recognition

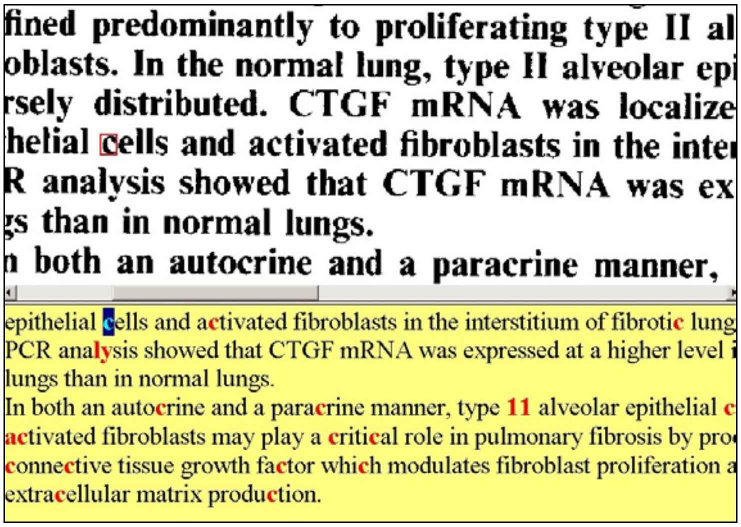
Two problems observed in production proved to be amenable to lexical analysis techniques. The first problem was the excessive number of highlighted characters (which were actually correct, but assigned a low confidence level by the OCR system, and hence highlighted on the screen.) The second problem was the large number of character errors in the detected affiliations field, a consequence of small font size and italic attribute in the printed text in that field. Both problems placed an additional burden on the reconcile operators to correct and verify the text. Two modules, developed to solve these problems and reduce the operator labor, exploit the specialized vocabulary found in biomedical journals. While the modules use different techniques, both employ specially selected lexicons to modify the OCR text that is presented to the reconcile operators.

8.1 Lexical analysis to reduce highlighted words

8.1.1 Problem Statement

The Prime Recognition OCR system was selected for its high rate of correctly recognized characters (high detection accuracy) and the very low number of incorrectly recognized characters that were assigned a high confidence value (low false positives). Confidence levels lie in a range between 1 and 9. Trading off the low percentage of false positives, we found that over 90% of words containing low confidence characters are actually correct, and that these characters should have been assigned a value of 9 by the OCR system. To draw the reconcile operators’ attention to characters that may need correction, all low confidence characters are highlighted in red on the reconcile workstation screen.

When these are mostly correct, the operators are unnecessarily burdened by having to examine and tab through them. Figure 8.1.1 shows a portion of the reconcile screen, with characters highlighted incorrectly, i.e., with the original confidence values from the OCR system.



fined predominantly to proliferating type II al
oblasts. In the normal lung, type II alveolar epi
rsely distributed. CTGF mRNA was localize
helial cells and activated fibroblasts in the inte
R analysis showed that CTGF mRNA was ex
gs than in normal lungs.
n both an autocrine and a paracrine manner,

epithelial cells and activated fibroblasts in the interstitium of fibrotic lung
PCR analysis showed that CTGF mRNA was expressed at a higher level i
lungs than in normal lungs.
In both an autocrine and a paracrine manner, type 11 alveolar epithelial c
activated fibroblasts may play a critical role in pulmonary fibrosis by pro
connective tissue growth factor which modulates fibroblast proliferation a
extracellular matrix production.

Figure 8.1.1

In this example, part of the bitmapped image of the abstract field is displayed at the top of the screen and the corresponding OCR output text is displayed at the bottom. Although all of the OCR text is correct in this example, many characters are highlighted in red. Our objective is to reduce this number of highlighted characters.

8.1.2 Approach

To reduce the number of (incorrectly) highlighted characters, we designed a module to automatically increase the confidence level of characters detected correctly by the OCR system. This module locates each word in the title and abstract fields that contains any low confidence characters, checks for the word in a lexicon and, if the word is found, changes the confidence of all its characters to 9, the highest value.

A study was undertaken to determine criteria (heuristic rules) for selecting words to be checked and a lexicon suitable for biomedical journal articles. The key element of the study was the creation of a ground truth dataset with which to compare lexicons and lookup criteria. The ground truth data consisted of 5,692 OCR output words containing low confidence characters extracted from journals already processed by the MARS system. Each of these words was compared to the corresponding word in the final, verified bibliographic record created by MARS to determine if the OCR word was correct or not. Candidate lexicons and lookup criteria were evaluated with the goal of removing low confidence values from ground truth words that were correct, while retaining the low confidence values for those words that were not correct. Removing low confidence values from correct words is the "benefit" of the module. Removing low confidence values from incorrect words is the potential "cost" of the module.

8.1.3 Experiments and results

Four candidate lexicons were created from various word lists maintained by the National Library of Medicine with the expectation that these would contain a preponderance of the biomedical words found in journal articles indexed in MEDLINE. The four lexicons and their combinations were tested along with several lookup criteria involving word length and character confidence levels. As expected, there was a tradeoff between benefit and cost. A large lexicon and no lookup restrictions removed low confidence values from over 90% of the OCR correct words (a 90% benefit), but also removed low confidence values from over 60% of the OCR incorrect words (a 60% cost). To ensure the integrity of the final text, it was considered on balance more important to minimize cost than to maximize benefit.

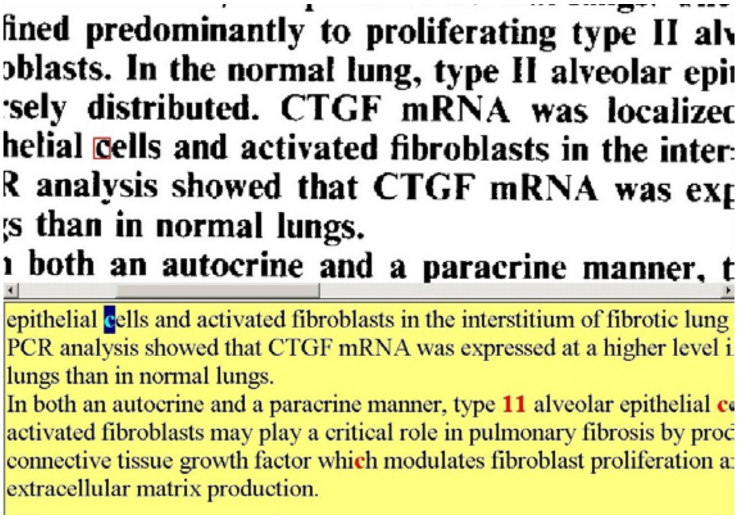
Three combinations of lexicons and lookup criteria resulted in acceptable costs of less than 0.5% and benefits greater than 40%. The final choice correctly removed low confidence values from 46% of the correct OCR words and incorrectly removed low confidence values from 0.4% of the incorrect OCR words. The selected lexicon consists of unique words derived from the 1997 editions of NLM's SPECIALIST Lexicon and UMLS Metathesaurus. There are two levels of lookup criteria: 1) Words less than four characters in length, or containing no alphabetic characters are not checked. 2) Words less than six characters in length are not checked if any of the confidence values are less than 7. All other words containing low confidence characters are compared to the lexicon. If the word is found, the confidence values for all the characters are changed to 9, the highest value.

8.1.4 Implementation

The lexicon checking module was implemented at two phases of the project, the first for the MARS-1 system and later for the MARS-2 system. For the MARS-1 system it was implemented as a console application, written in C and developed in the Microsoft Visual C++ development environment. The selected lexicon was compressed and organized into a special dictionary format for fast searching using the commercially available software, Visual Speller. In production, it was found that in the MARS-1 system, lexicon checking reduced the highlighted words on average from approximately 14% of the words presented for verification at the reconcile workstation to approximately 6.5%. This 50% reduction in highlighted words resulted in a 4% increase in production rate, and was reported in the literature[21].

For the MARS-2 system another module was developed to implement the algorithm described above. This module, a console application called Confidence Edit, is written in C++ in the Microsoft Visual Studio development environment. As is the case for all MARS-2 modules, it reads data records from the system database and creates new records with edited (increased) confidence values. The lexicon is also stored in a database table. When Confidence Edit starts, it loads the lexicon into a ternary search tree in memory. The memory structure is compact, supports very fast lookup and presents no load on the database server. In the MARS-2 system, Confidence Edit has reduced highlighted words in the abstract field from approximately 7.5% to approximately 4.3%, using the same lexicon and lookup criteria as used by the original MARS-1 module. Figure 8.2 illustrates the effect of processing by Confidence Edit on the same document shown in Figure 8.1. Most of the characters are no longer highlighted.

Improvements made since Confidence Edit was placed in production included an addition of 9,386 words to the lexicon. These were obtained by extracting all the words found in the verified and corrected abstracts from over 27,000 journals (= 230,000 articles) processed by MARS-1 and MARS-2 from May 1997 to April 2001, and using the frequency of occurrence of each word during that period. New words occurring at a frequency of 50 or more were added to the lexicon. Remaining words that occurred at least twice were checked against nine electronic dictionaries that are available to NLM. If the word was found in Dorland's Medical Dictionary, in the Oxford Medical Dictionary, or in at least six other dictionaries, it was added to the lexicon. When the new lexicon was tested with the original ground truth data, a 4% improvement in benefit with no increase in cost was measured. Similar statistics were found with a selected set of test journals. Using the expanded lexicon, Confidence Edit has reduced the percentage of highlighted words in the abstract field to approximately 3.5%, a modest improvement.



fined predominantly to proliferating type II alv
oblasts. In the normal lung, type II alveolar epi
rsely distributed. CTGF mRNA was localizec
helial cells and activated fibroblasts in the inter
R analysis showed that CTGF mRNA was exp
gs than in normal lungs.
n both an autocrine and a paracrine manner, t

epithelial cells and activated fibroblasts in the interstitium of fibrotic lung
PCR analysis showed that CTGF mRNA was expressed at a higher level i
lungs than in normal lungs.
In both an autocrine and a paracrine manner, type 11 alveolar epithelial c
activated fibroblasts may play a critical role in pulmonary fibrosis by proc
connective tissue growth factor which modulates fibroblast proliferation a
extracellular matrix production.

Figure 8.1.2

8.2 Lexical analysis to improve recognition of Affiliations

8.2.1 Problem Statement

As noted previously, although the commercial OCR system used by MARS performs well in general, accuracy is often poor for small fonts or italic characters. In particular, authors' affiliations frequently appear as small and/or italic characters, resulting in many incorrect characters in the affiliation field. Consequently, the final check and correction of the affiliation field requires a disproportionate amount of human labor compared to other fields extracted by our automated system. We observed that for about one in five affiliations, there were so many highlighted words that the operators preferred to type the entire affiliation rather than examine and correct each word. Not only was this time consuming, but also represented a potential source of error in the completed bibliographic record because fields manually entered at the reconcile workstation are not double-keyed as at the edit stage. In this section we describe the experiments that led to the design and implementation of the PatternMatch module that corrects words in the affiliation field.

8.2.2 Approach

Words that frequently appear in the affiliation field in biomedical journals are drawn from a relatively small vocabulary that denote institutions and their divisions, such as University and Department, the various branches of medicine and biology, such as Pathology and Biophysics, and names of cities, states and countries. A study was undertaken to determine if partial string matching or other matching techniques could exploit the limited vocabulary to reliably find the correct word from a lexicon of affiliation words given an OCR output word containing low confidence characters.

As in the previously described problem, a key component of this study was a ground truth dataset to compare matching techniques and lexicons. Words containing low confidence characters and the confidence values for all characters were extracted from the OCR output text in the affiliations field of over five thousand journal articles processed by the MARS system. Human operators selected the corresponding correct word from the affiliation fields of the completed bibliographic records. After words that contained no alphabetic characters were removed, the ground truth data consisted of over 20 thousand triplets, where a triplet consists of an OCR output word, the confidence values for the characters in the word and the correct word. Over 60% of the OCR words in the ground truth set are correct.

Correct words and a count of their occurrences were extracted from the final, corrected affiliation field of approximately 230,000 journal articles that had already been processed by the MARS system. This set of journal articles is different from the set used to create the ground truth data. There were 96,982 unique words of 2 or more characters that occurred one or more times in this historical data. This list of words is the basis for candidate lexicons of affiliation words.

8.2.3 Experiments

Six matching techniques[42] were tested using the ground truth data and the complete lexicon of affiliation words. The goal of testing was to find a technique to achieve a high match rate, a low false positive rate and fast processing. Most of the techniques return more than one potential match from a lexicon. If the correct word is among the list of returned words, it is considered a match. If no words are returned, it is considered no match. If words are returned, but none of them are the correct word, it is considered a false positive. The six techniques tested are:

Whole Word Matching. This technique compares the entire OCR output word with each word in the lexicon. Either one word is returned, or none is. Because over 60% of the OCR words in the ground truth set are correct, the match rate was reasonably high. However, when using the complete lexicon, the rate of false positives was also high because a single OCR error or omission can result in a word that is found in the lexicon. For example, several non-English variations of the word University are included in the lexicon, including Universit, Universita, Universitaire, Universitat, and Universite. If the OCR word is "Universit", whole word matching will find "Universit" even though the actual word was "University" or one of its other variations.

Partial Matching with wild card letters. In this technique, a match is sought with the "wild card" character '.' (a period) substituted for one or more of the low confidence characters in the OCR output word. Thus Partial Matching can find words in the reference dictionary where the OCR word has one or more character errors, but whose length is correct. For example, if we have an OCR word, Deparlmemt, with confidence values, 9699878956, a match would be found for Depar.me.t or D.par.me.t, but not for D.parlme.t or D.par.memt. For the same reasons as for Whole Word Matching, the false positive rate was high. In addition, the method does not find a correct match if the number of characters in the OCR word is incorrect.

Near-neighbor Matching. This technique finds all of the words in the lexicon that are within a given Hamming distance of the OCR word. Hamming distance is a measure of the difference between two finite strings of characters, expressed as the number of characters that need to be changed to obtain one from the other. For example, “Deparlmemt” and “Department” have a Hamming distance of two, whereas "Butter" and "ladder" have a Hamming distance of four. A high match rate could be achieved by specifying a large Hamming distance, but this also resulted in a high false positive rate. Near-neighbor Matching also fared poorly when the OCR word length was incorrect.

Soundex Matching. Soundex matching finds words in the lexicon that are phonetically similar to the OCR output word. This technique proved to have a number of difficulties in overcoming character substitutions caused by the OCR system. For example, the word Department would often be interpreted as Deparlment by the OCR engine. These two words are phonetically quite different since the letters ‘t’ and ‘l’ do not have similar sounds, and the correct word, Department, would not be returned.

Bi-gram Search. This technique was adapted from software developed inhouse to suggest spelling alternatives for online Library clients. A bi-gram is a pair of adjacent characters in the word being analyzed. For example, Department contains nine bi-grams: De, ep, pa, ar, rt, tm, me, en, nt. For bi-gram searches, each word in the lexicon is searched for all possible bi-grams in the OCR word. Lexicon words containing multiple bi-grams in the OCR word are possible matches. Long OCR words can result in a large number of possible matches. The bi-gram prunes out unlikely candidate words through an algorithm that considers lexicon word length, OCR word length and the number of matching bi-grams. Bi-gram searches were less sensitive to OCR word length than some other techniques and resulted in a relatively high match rate. The false positive rate was also high, because one or two incorrect characters in the OCR word could result in many incorrect possible matches.

Probability Matching. Probability Matching[43] was developed inhouse specifically to address the problem of poor OCR recognition of words in the affiliation field. The first step of this technique compares the OCR output word to every word in the lexicon using an edit distance based on OCR character substitution frequencies, and assigns a confidence value to each lexicon word based on the probability that the OCR word will be produced when the true word is the lexicon word. Each word receives a score that is the product of the calculated confidence and the frequency of occurrence in the lexicon. Words are then ranked according to this score, and a specified number of highest-ranking words are returned.

Probability Matching achieved the highest match rate, lowest false positive rate, and slowest processing time of the techniques tested. Because it compares the OCR word with every word in the lexicon, it can take up to 1 second per word depending on word length, computer speed and lexicon size, with larger lexicons producing better matching and slower processing.

Initial experiments suggested further study toward an acceptable multi-stage process in which "easy" words are matched reliably by a faster process, and the remaining words are matched using Probability Matching. Combinations of matching techniques and lexicon sizes were tested in an effort to reduce the false positive rate and the processing time while maintaining the high match rates that had been observed.

8.2.4 Results

The final choice was a cascaded matching process that capitalizes on the fact that over half of the OCR words are correct. It was found that trimming the lexicon to include only the most frequently occurring words significantly reduced the false positive rate of Whole Word Matching. Probability Matching with a larger lexicon for words not found by Whole Word Matching then yielded acceptable overall results with less impact on average processing time.

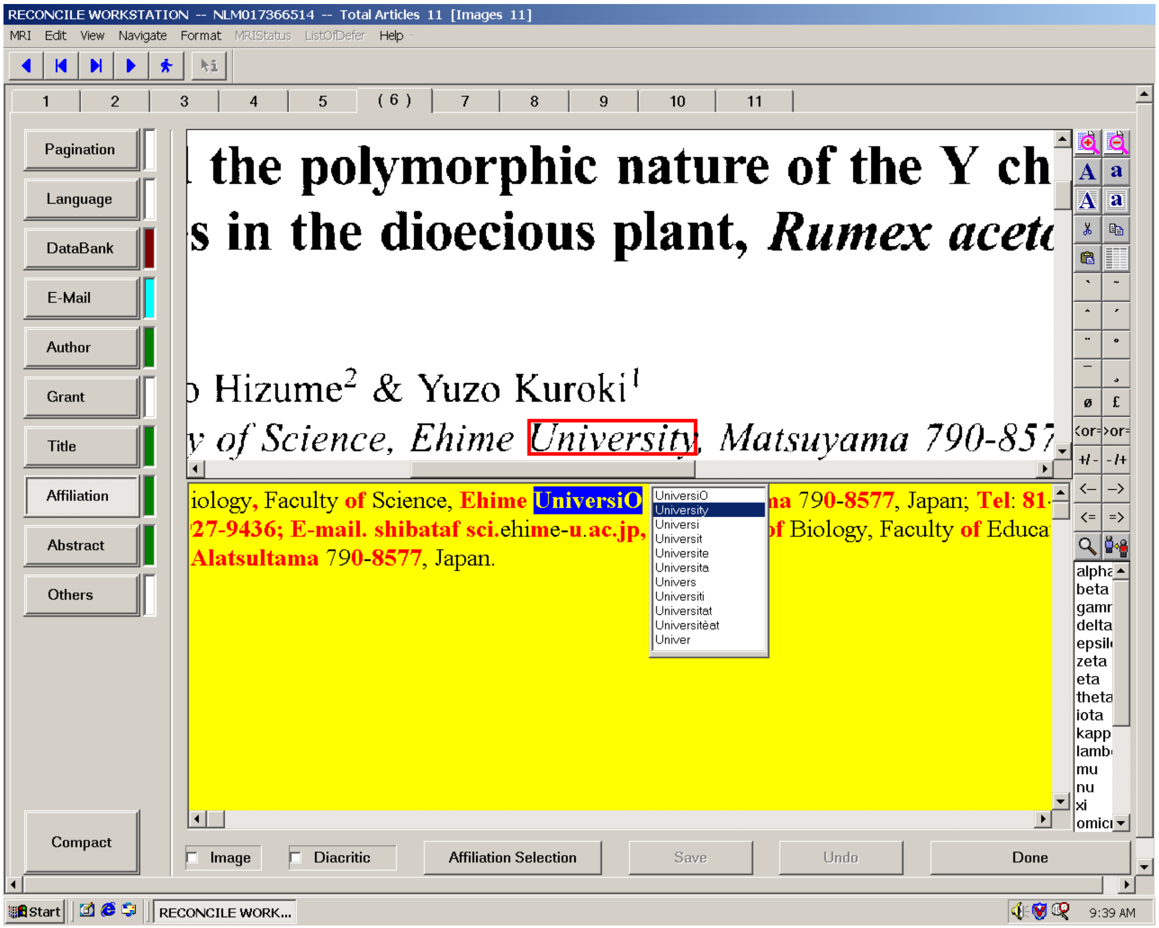
The first step in our cascade matching is Whole Word Matching with a lexicon of 1,948 affiliation words with an occurrence of 100 or more in the historical data. Step 1 correctly matches 45% of the ground truth data, with a false positive rate of 1.4%. The second step is Probability Matching with the entire lexicon of 43,030 affiliation words with an occurrence of 2 or more in the historical data. Step 2 correctly matches 77% of the remaining 55% of the ground truth data, with a false positive rate of 16.7%. The overall performance of cascade matching was a match rate of 86% (with the correct word ranked highest for 81% of the words), a false positive rate of 11% and an average processing time of approximately 250 ms on a 500 MHz Pentium III. The still high false positive rate has implications for the way that potential substitute words are presented to the reconcile operator.

8.2.5 Implementation

Word matching for low confidence words in the affiliation field was implemented through two software development efforts[42]. A separate console program called PatternMatch was developed to automatically parse low confidence OCR words from the identified affiliation field, submit the words to the cascade matching algorithm for possible correct words, and, if any words are returned, insert those words into the affiliation text following the original OCR word and specially tagged for processing by the Reconcile workstation. PatternMatch was developed in C++ in the Microsoft Visual Studio development environment. In addition to the MARS database for input and output records, PatternMatch requires three files: the large and small lexicons and the character substitution frequency matrix used by Probability Matching.

Additional software was developed for reconcile to support the interpretation of the special tags placed in the affiliation text by PatternMatch and the display of the word choices to the reconcile operator. An example of the reconcile application screen is shown in Figure 8.2.1. In this instance, the original OCR output word containing low confidence characters, seen in the lower half of the figure, is *UniversiO*. The upper half of the screen displays the scanned image with a red box around the image corresponding to the word highlighted in the lower half of the screen. In this example the word matching process found 10 words that could possibly match UniversiO. These are presented to the operator in a drop down list. The first word in the list (*UniversiO)* is the original OCR word, the default highlighted word (*University*) is the highest ranked word match, the third word in the list (*Universi*) is the second highest ranked word match, and so on. The reconcile operator has the option to hit escape and leave the original word highlighted, hit return to substitute the original OCR word with the highest ranked word,, or select any of the words in the list using the mouse or keyboard and hit return. The ease of selection is relative to the need for correction: selecting the original OCR word or the first candidate word in the match list is accomplished with a single keystroke. These cases account for approximately 90% of the words containing low confidence characters.

PatternMatch was placed in operation in December 2000, and the reconcile software to support word selection was introduced in February 2001. In the first four months of operation, operators selected the first match choice 80.8% percent of the time, selected one of the other match choices 8.8% of the time, selected no word from the list 6.7% of the time, and the original OCR word 3.7% of the time.



l the polymorphic nature of the Y ch
s in the dioecious plant, *Rumex acet*

ɔ Hizume[2] & Yuzo Kuroki[1]
y of Science, Ehime University, Matsuyama 790-857

iology, Faculty of Science, Ehime UniversiO na 790-8577, Japan; Tel: 81-
27-9436; E-mail. shibataf sci.ehime-u.ac.jp, of Biology, Faculty of Educa
Alatsultama 790-8577, Japan.

Figure 8.2.1 PatternMatch output for reconcile operator

8.2.6 Incremental Improvements

Two modifications to the subsystem for affiliation word matching were implemented in the summer of 2001. A new compilation of affiliation words was generated from a larger set of verified MARS records to generate more complete lexicons for the PatternMatch program. In addition, PatternMatch was modified to more accurately handle words containing diacritics.

9 Operator workstation design

In this section we describe the three principal types of operator workstations, for scanning, editing and reconciling. In all cases, off-the-shelf hardware is used.

9.1 Scan workstation

The scanners in production are mid-range devices manufactured by Fujitsu or Ricoh. These devices are controlled by an inhouse application software called Scan. The primary task for this software is to enable an operator to *scan* a page of an article to produce a TIFF image, and *insert, delete* or *replace* a page image. This software also allows the operator to initiate the workflow for a journal issue (i.e., by entering the MRI number to identify the journal issue to be processed) in case the first stage in MARS, CheckIn, fails or a supervisor is unable to initiate that process for any reason. This is a feature that contributes to overall system reliability. In addition, Scan requires an operator to check the quality of the image resulting from scanning; the operator may zoom into any part of the image to ensure that the page has been scanned correctly. Finally, in case a journal is sent back from downstream processes for rescanning, the Scan software identifies the pages to be rescanned for the operator.

The Scan software is written in C++ and compiled in Microsoft Visual C++ (6.0). The GUI design is based on the AppWizard in MFC, using the option of Single Document Interface (SDI). The parallel process is not required since the operator works on only one page in a journal issue at a time.

The software uses ActiveX to control the scanner through a Kofax controller. The TIFF images are displayed, magnified, rotated and scaled through another ActiveX control provided by Eastman Kodak Image software. Communications with the MARS database are accomplished through RogueWave functions. These are shown in the schematic below.

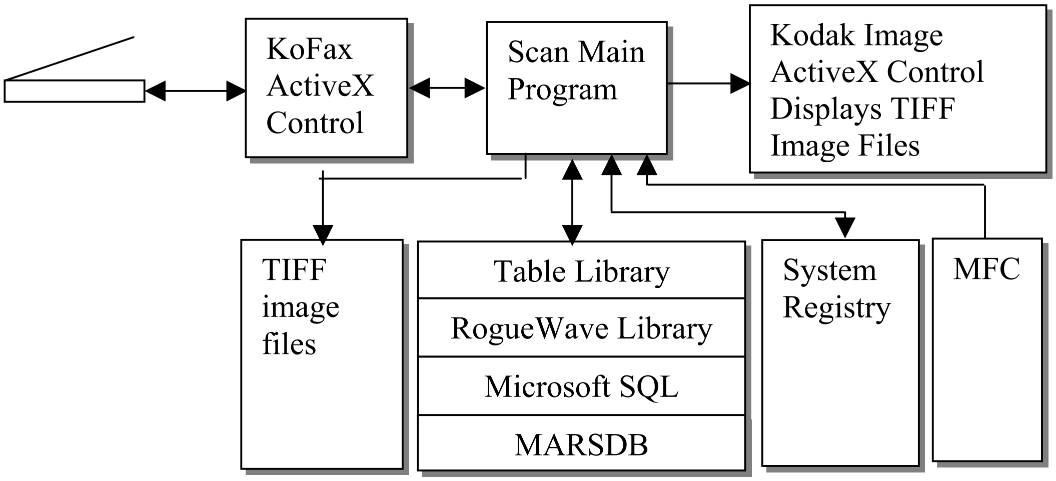
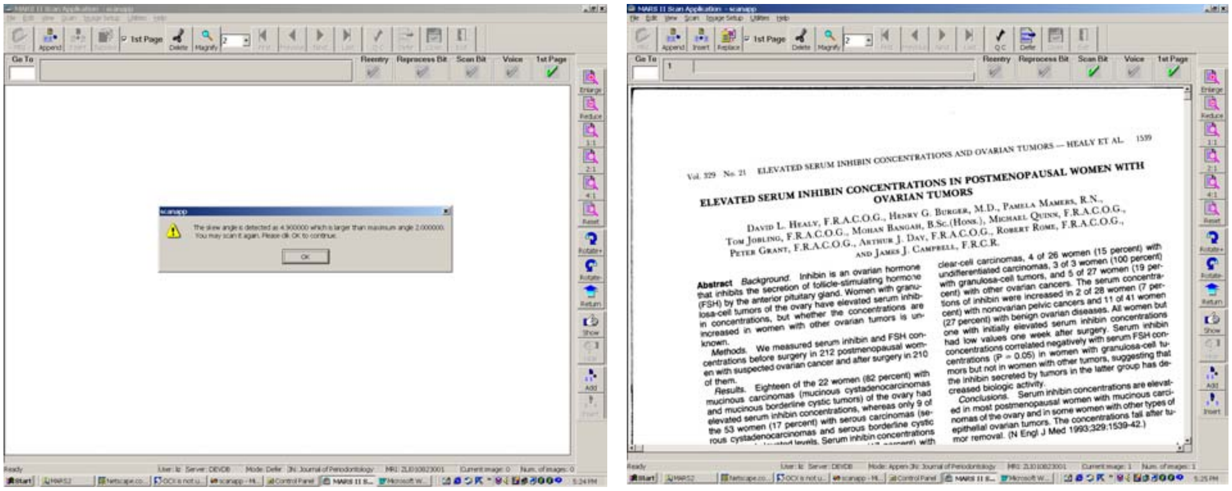


Figure 9.1.1 Scan software schematic

The Scan program implements real-time communications between the scanner and the MARS database. Should this communication fail, Scan alerts the operator immediately. Scan creates records in the WIP, Page and ProcessTime tables in the database. In the WIP table, it records the journal issue identification (the MRI number barcode scanned in by the operator), time the record is archived, time the journal issue is scanned, location of the TIFF images, total number of images in the issue, the operator ID, the type of scanner, and other data. In the Page table, it records a unique number identifying each page (PageID), scan density (dpi), the height and width of the page in pixels, and whether the scanned page is the first or second page of the article (the second page is scanned only if the abstract continues on to this page). In the ProcessTime table, it records the start and end time for scanning a page, and whether scanning is mouse/keyboard driven or speech controlled.

The Scan software has a quality control (QC) function requiring the operator to view the image before exiting the application. To give the operator a quick indication of quality, we have provided a skew detection capability, skew being a factor in poor image quality: the operator is alerted if the skew exceeds a preset threshold, as shown in Figure 9.1.2.



The skew angle is detected as 4.900000 which is larger than maximum angle 2.000000.
You may scan it again. Please clk OK to continue.

Vol. 329 No. 21 ELEVATED SERUM INHIBIN CONCENTRATIONS AND OVARIAN TUMORS — HEALY ET AL. 1539

ELEVATED SERUM INHIBIN CONCENTRATIONS IN POSTMENOPAUSAL WOMEN WITH OVARIAN TUMORS

David L. Healy, F.R.A.C.O.G., Henry G. Burger, M.D., Pamela Mamers, R.N.,
Tom Jobling, F.R.A.C.O.G., Mohan Bangah, B.Sc.(Hons.), Michael Quinn, F.R.A.C.O.G.,
Peter Grant, F.R.A.C.O.G., Arthur J. Day, F.R.A.C.O.G., Robert Rome, F.R.A.C.O.G.,
and James J. Campbell, F.R.C.R.

Abstract *Background.* Inhibin is an ovarian hormone that inhibits the secretion of follicle-stimulating hormone (FSH) by the anterior pituitary gland. Women with granulosa-cell tumors of the ovary have elevated serum inhibin concentrations, but whether the concentrations are increased in women with other ovarian tumors is unknown.

Methods. We measured serum inhibin and FSH concentrations before surgery in 212 postmenopausal women with suspected ovarian cancer and after surgery in 210 of them.

Results. Eighteen of the 22 women (82 percent) with mucinous carcinomas (mucinous cystadenocarcinomas and mucinous borderline cystic tumors) of the ovary had elevated serum inhibin concentrations, whereas only 9 of the 53 women (17 percent) with serous carcinomas (serous cystadenocarcinomas and serous borderline cystic

clear-cell carcinomas, 4 of 26 women (15 percent) with undifferentiated carcinomas, 3 of 3 women (100 percent) with granulosa-cell tumors, and 5 of 27 women (19 percent) with other ovarian cancers. The serum concentrations of inhibin were increased in 2 of 28 women (7 percent) with nonovarian pelvic cancers and 11 of 41 women (27 percent) with benign ovarian diseases. All women but one with initially elevated serum inhibin concentrations had low values one week after surgery. Serum inhibin concentrations correlated negatively with serum FSH concentrations (P = 0.05) in women with granulosa-cell tumors but not in women with other tumors, suggesting that the inhibin secreted by tumors in the latter group has decreased biologic activity.

Conclusions. Serum inhibin concentrations are elevated in most postmenopausal women with mucinous carcinomas of the ovary and in some women with other types of epithelial ovarian tumors. The concentrations fall after tumor removal. (N Engl J Med 1993;329:1539-42.)

Figure 9.1.2 An alert to the operator (left); the actual image (right).

The workstation using the Scan software may be controlled conventionally by mouse/keyboard as well as by speech recognition. The design of the speech-enabled workstation, and the tradeoffs considered among different speech recognition approaches, are reported in the literature[58].

9.2 Edit workstation

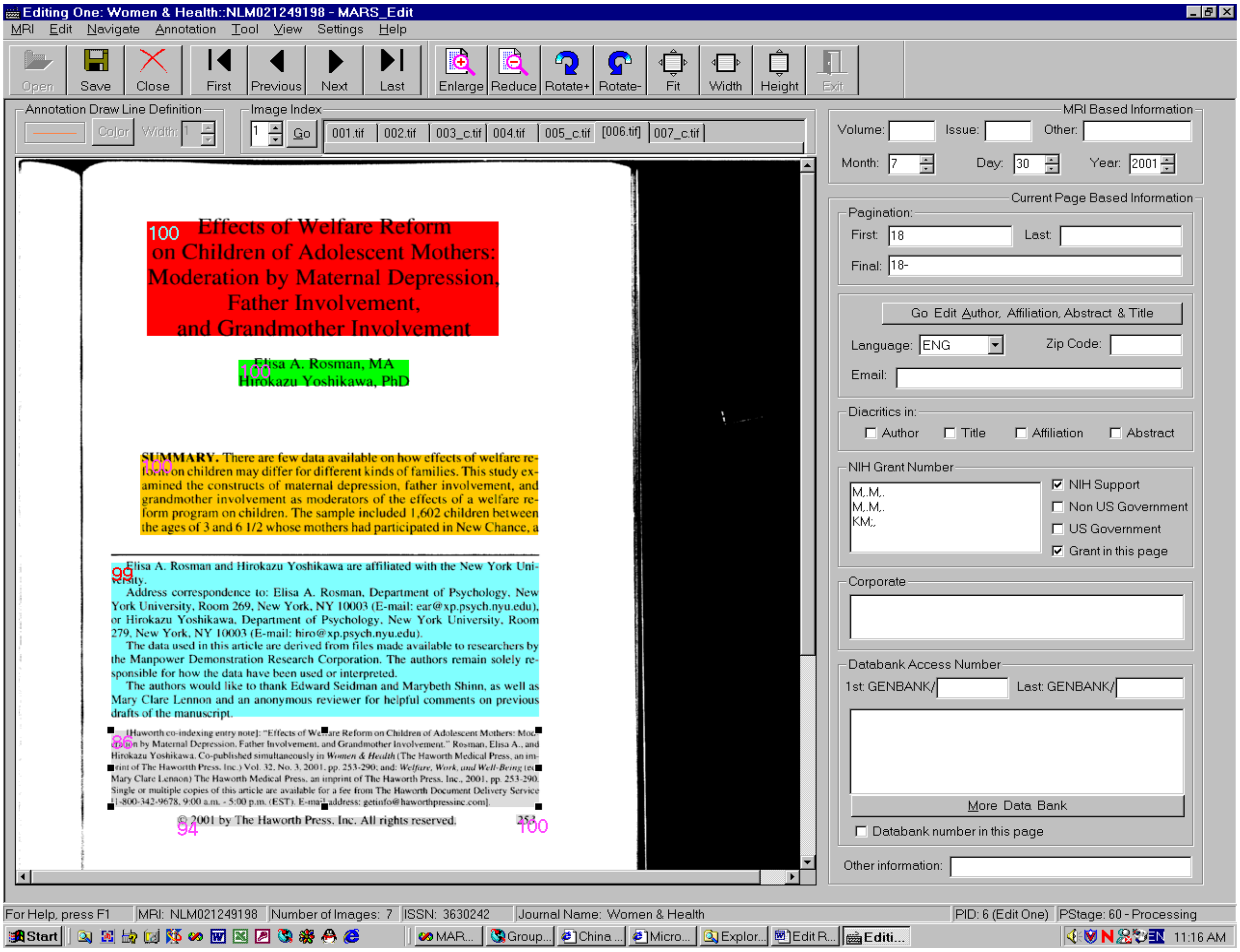
While our goal is to automate data extraction from the scanned journals as much as possible, at any point in the life cycle of the MARS system there will be a need for some manual data entry for the fields not automatically extracted. One reason is that some data appears in pages other than the first page of the article, the only page that is usually scanned. (If the abstract continues on to the article’s second page, this page is also scanned, but this occurs infrequently.) Examples are: NIH grant numbers and databank accession numbers. Another reason is that the OCR system does not reliably detect very small or italicized characters, such as in the affiliation field; in this situation, the edit operator might choose to simply type in the text. Furthermore, the MARS-2 algorithms handle only the “compliant” journals, while a significant, though decreasing, portion of the journal collection remains noncompliant. The edit workstation is designed for manual data entry, the name reflecting the combined actions of the editor and keyboard operator in the MARS-1 system.

The edit workstation takes advantage of upstream processes (OCR, autozoning and autolabeling), provides an interface for data entry, and allows the operator to correct errors in zoning and labeling before passing the data on to subsequent processes.

In order to minimize human errors at the edit stage, data entry is double keyed, i.e., two different operators produce two versions of the data for the same article: one version in Edit_One, another version in Edit_Two. The two versions of the data are differenced by a daemon process called Diff so that the reconcile operator may clearly detect any differences and select the correct version.

As shown in the workstation GUI in Figure 9.2.1, the edit operator is required to enter the pagination and language fields, the latter set to English as the default option. All other fields are optional, entered only if necessary. The left window of the workstation screen displays the TIFF page image including the (color coded) results of autozoning and autolabeling performed by upstream processes, so that the operator can identify and correct an incorrectly labeled zone, to ensure that errors do not propagate to downstream processes such as Confidence Edit, Reformat and Reconcile.

In addition to showing the zones and labels automatically done by daemons, the system also displays the *percentage of high confidence OCR output characters* (confidence level of 9) for each zone. These items of information displayed on the bitmapped image serve as a DoItAgain feature, enabling the edit operator to request the Admin operator to order redoing earlier processes such as Scan, OCR, zoning and labeling which may have produced errors resulting from poor scanning, unacceptably high misinterpretation by the OCR system, or incorrect zoning or labeling.


Editing One: Women & Health::NLM021249198 - MARS_Edit
MRI Edit Navigate Annotation Tool View Settings Help
Open
Save
Close
First
Previous
Next
Last
Enlarge
Reduce
Rotate+
Rotate-
Fit
Width
Height
Exit
Annotation Draw Line Definition
Color
Width:
Image Index
Go
001.tif 002.tif 003_c.tif 004.tif 005_c.tif [006.tif] 007_c.tif
Go Edit Author, Affiliation, Abstract & Title
More Data Bank
Start
MAR...
Group...
China ...
Micro...
Explor...
Edit R...
Editi...
11:16 AM

Width: 1

Image Index: 1

100 Effects of Welfare Reform
on Children of Adolescent Mothers:
Moderation by Maternal Depression,
Father Involvement,
and Grandmother Involvement

100 Elisa A. Rosman, MA
Hirokazu Yoshikawa, PhD

100 **SUMMARY.** There are few data available on how effects of welfare reform on children may differ for different kinds of families. This study examined the constructs of maternal depression, father involvement, and grandmother involvement as moderators of the effects of a welfare reform program on children. The sample included 1,602 children between the ages of 3 and 6 1/2 whose mothers had participated in New Chance, a

99 Elisa A. Rosman and Hirokazu Yoshikawa are affiliated with the New York University.

Address correspondence to: Elisa A. Rosman, Department of Psychology, New York University, Room 269, New York, NY 10003 (E-mail: ear@xp.psych.nyu.edu), or Hirokazu Yoshikawa, Department of Psychology, New York University, Room 279, New York, NY 10003 (E-mail: hiro@xp.psych.nyu.edu).

The data used in this article are derived from files made available to researchers by the Manpower Demonstration Research Corporation. The authors remain solely responsible for how the data have been used or interpreted.

The authors would like to thank Edward Seidman and Marybeth Shinn, as well as Mary Clare Lennon and an anonymous reviewer for helpful comments on previous drafts of the manuscript.

86 [Haworth co-indexing entry note]: "Effects of Welfare Reform on Children of Adolescent Mothers: Moderation by Maternal Depression, Father Involvement, and Grandmother Involvement." Rosman, Elisa A., and Hirokazu Yoshikawa. Co-published simultaneously in *Women & Health* (The Haworth Medical Press, an imprint of The Haworth Press, Inc.) Vol. 32, No. 3, 2001, pp. 253-290; and: *Welfare, Work, and Well-Being* (ed Mary Clare Lennon) The Haworth Medical Press, an imprint of The Haworth Press, Inc., 2001, pp. 253-290. Single or multiple copies of this article are available for a fee from The Haworth Document Delivery Service [1-800-342-9678, 9:00 a.m. - 5:00 p.m. (EST). E-mail address: getinfo@haworthpressinc.com].

94 © 2001 by The Haworth Press, Inc. All rights reserved. 253 100

MRI Based Information
Volume: Issue: Other:
Month: 7 Day: 30 Year: 2001

Current Page Based Information
Pagination:
First: 18 Last:
Final: 18-

Language: ENG Zip Code:
Email:

Diacritics in:
□ Author □ Title □ Affiliation □ Abstract

NIH Grant Number
M,.M,.
M,.M,.
KM;,
☑ NIH Support
□ Non US Government
□ US Government
☑ Grant in this page

Corporate

Databank Access Number
1st: GENBANK/ Last: GENBANK/
□ Databank number in this page

Other information:

For Help, press F1 | MRI: NLM021249198 | Number of Images: 7 | ISSN: 3630242 | Journal Name: Women & Health | PID: 6 (Edit One) | PStage: 60 - Processing

Figure 9.2.1 Edit workstation GUI

The operator may also defer the process, or set an error flag at any time as needed, providing the flexibility for trouble-shooting and production scheduling.

The Edit software, running under Windows 2000, is developed using Visual C++ 6.0 and MFC 6.0. It uses Kodak Imaging Professional 2.5 to handle all image related functionality, and RogueWave Tools and DBTools 3.20 to implement client communications with the SQL database server.

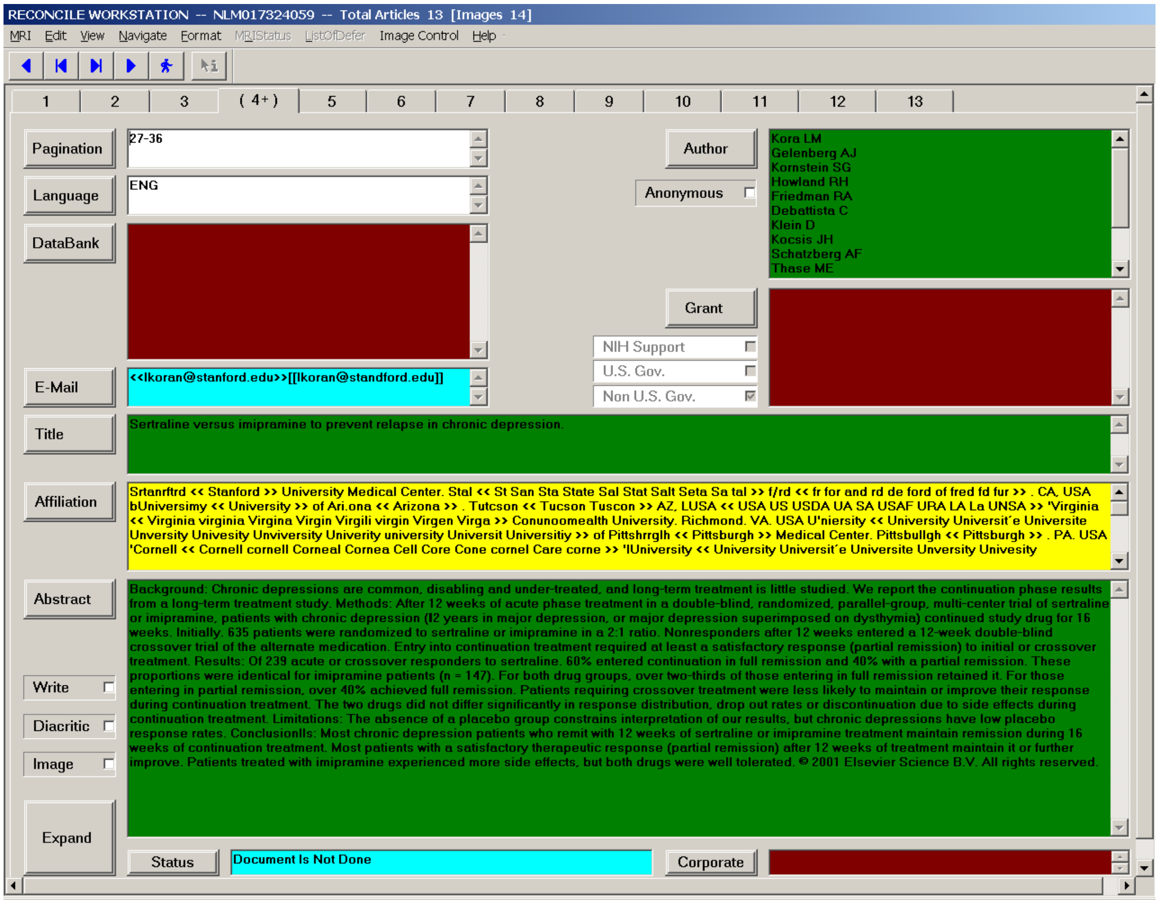
The Edit software is equipped to assist in system performance evaluation, by automatically recording the time taken by the operator to key in data (entered in the PerformanceData table in the database), and the processing time (entered in the ProcessTime table). It also incorporates error handling components to automatically detect and handle database and developer defined errors. When an error condition is triggered, the Edit software warns the operator and allows the entry of more information. This information is automatically recorded and handled by the Admin module.

The design of this workstation also seeks to minimize delays caused by data and image transfer over the network. For example, after completing data entry for one of the articles in a journal issue, the operator clicks the page tab to go to the next TIFF image. At this point, the data for the current article must be written to the database, and the information related to the next article must be retrieved from the database and its page image from the file server for display on the screen. The database I/O and image retrieval from the file server and its transfer over the LAN are all time consuming. To reduce the effective delay perceived by the operator, the Edit software reads the database information related to all of the articles in a particular journal issue into the workstation memory at the beginning of the process, so that this data is available immediately as the operator moves from one article to the next.

9.3 Reconcile workstation

The purpose of the reconcile workstation is to enable an operator to check the accuracy of the bibliographic data extracted by the automated processes, as well as that entered manually by the Edit operator. Any errors are corrected at this stage before the citation is uploaded to the DCMS database.

The general view in the reconcile workstation's GUI gives an overall picture of the bibliographic data captured (Figure 9.3.1). Prior to the operator verifying the contents of the bibliographic fields, the field windows are highlighted by background color: green for fields created by the automated processes, cyan for those entered manually, yellow for those created by combining the outputs of both automated and manual processes, and red if no text appears in the window. In the example shown, the windows for NIH Grant Numbers and Databank Accession Numbers appear in red, since these data were not entered by the edit operator earlier in the workflow. (These fields are not captured automatically since they could appear anywhere in the article and not just on the scanned page.) Once the operator verifies the fields, the windows turn white, as shown for Pagination and Language in this example.



Pagination: 27-36

Language: ENG

DataBank:

E-Mail: <<lkoran@stanford.edu>>[[lkoran@standford.edu]]

Author:
Kora LM
Gelenberg AJ
Kornstein SG
Howland RH
Friedman RA
Debattista C
Klein D
Kocsis JH
Schatzberg AF
Thase ME

Anonymous □

Grant:

NIH Support □
U.S. Gov. □
Non U.S. Gov. ☑

Title: Sertraline versus imipramine to prevent relapse in chronic depression.

Affiliation: Srtanrftrd << Stanford >> University Medical Center. Stal << St San Sta State Sal Stat Salt Seta Sa tal >> f/rd << fr for and rd de ford of fred fd fur >> . CA, USA bUniversimy << University >> of Ari.ona << Arizona >> . Tutcson << Tucson Tuscon >> AZ, LUSA << USA US USDA UA SA USAF URA LA La UNSA >> 'Virginia << Virginia virginia Virgina Virgin Virgili virgin Virgen Virga >> Conunoomealth University. Richmond. VA. USA U'niersity << University Universit´e Universite Unversity Univesity Unviversity Univerity university Universit Universitiy >> of Pittshrrglh << Pittsburgh >> Medical Center. Pittsbullgh << Pittsburgh >> . PA. USA 'Cornell << Cornell cornell Corneal Cornea Cell Core Cone cornel Care corne >> 'IUniversity << University Universit´e Universite Unversity Univesity

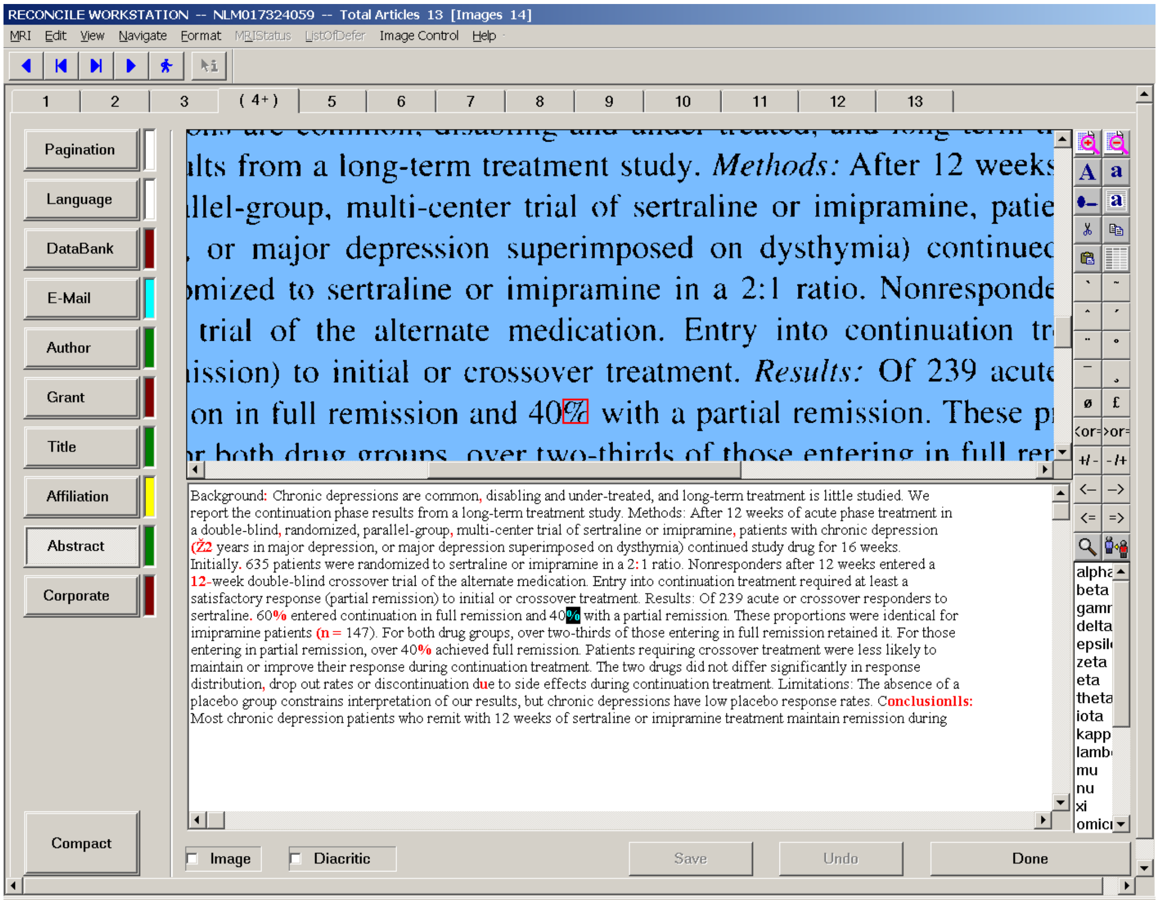
Abstract: Background: Chronic depressions are common, disabling and under-treated, and long-term treatment is little studied. We report the continuation phase results from a long-term treatment study. Methods: After 12 weeks of acute phase treatment in a double-blind, randomized, parallel-group, multi-center trial of sertraline or imipramine, patients with chronic depression (I2 years in major depression, or major depression superimposed on dysthymia) continued study drug for 16 weeks. Initially. 635 patients were randomized to sertraline or imipramine in a 2:1 ratio. Nonresponders after 12 weeks entered a 12-week double-blind crossover trial of the alternate medication. Entry into continuation treatment required at least a satisfactory response (partial remission) to initial or crossover treatment. Results: Of 239 acute or crossover responders to sertraline. 60% entered continuation in full remission and 40% with a partial remission. These proportions were identical for imipramine patients (n = 147). For both drug groups, over two-thirds of those entering in full remission retained it. For those entering in partial remission, over 40% achieved full remission. Patients requiring crossover treatment were less likely to maintain or improve their response during continuation treatment. The two drugs did not differ significantly in response distribution, drop out rates or discontinuation due to side effects during continuation treatment. Limitations: The absence of a placebo group constrains interpretation of our results, but chronic depressions have low placebo response rates. ConclusionlIs: Most chronic depression patients who remit with 12 weeks of sertraline or imipramine treatment maintain remission during 16 weeks of continuation treatment. Most patients with a satisfactory therapeutic response (partial remission) after 12 weeks of treatment maintain it or further improve. Patients treated with imipramine experienced more side effects, but both drugs were well tolerated. © 2001 Elsevier Science B.V. All rights reserved.

Status: Document Is Not Done

Corporate:

Figure 9.3.1 General view for all bibliographic fields in an article.

The GUI for this workstation also gives a split view (Figure 9.3.2) displaying both the image and the corresponding text to allow the operator to verify the text against the scanned image. Low confidence characters are highlighted in red on the text to draw the operator's attention to them.



ılts from a long-term treatment study. *Methods:* After 12 weeks
ıllel-group, multi-center trial of sertraline or imipramine, patie
, or major depression superimposed on dysthymia) continued
ɔmized to sertraline or imipramine in a 2:1 ratio. Nonresponde
trial of the alternate medication. Entry into continuation tr
ıission) to initial or crossover treatment. *Results:* Of 239 acute
on in full remission and 40% with a partial remission. These p

Background: Chronic depressions are common, disabling and under-treated, and long-term treatment is little studied. We report the continuation phase results from a long-term treatment study. Methods: After 12 weeks of acute phase treatment in a double-blind, randomized, parallel-group, multi-center trial of sertraline or imipramine, patients with chronic depression (Ž2 years in major depression, or major depression superimposed on dysthymia) continued study drug for 16 weeks. Initially. 635 patients were randomized to sertraline or imipramine in a 2:1 ratio. Nonresponders after 12 weeks entered a 12-week double-blind crossover trial of the alternate medication. Entry into continuation treatment required at least a satisfactory response (partial remission) to initial or crossover treatment. Results: Of 239 acute or crossover responders to sertraline. 60% entered continuation in full remission and 40% with a partial remission. These proportions were identical for imipramine patients (n = 147). For both drug groups, over two-thirds of those entering in full remission retained it. For those entering in partial remission, over 40% achieved full remission. Patients requiring crossover treatment were less likely to maintain or improve their response during continuation treatment. The two drugs did not differ significantly in response distribution, drop out rates or discontinuation due to side effects during continuation treatment. Limitations: The absence of a placebo group constrains interpretation of our results, but chronic depressions have low placebo response rates. Conclusionlls: Most chronic depression patients who remit with 12 weeks of sertraline or imipramine treatment maintain remission during

Figure 9.3.2 Split view shows both the bitmapped image and corresponding text. Low confidence characters are highlighted in the text window.

The reconcile workstation provides the operator several additional functions: the operator may redefine a field labeled incorrectly; activate a standalone OCR system to extract the field contents through the image or, alternatively, type in the text; if a page image is missing or duplicated, the operator may insert the missing page, or delete a duplicate; and if there are 'invalid' characters, the Reconcile software will convert them to the form required in MEDLINE.

Many of the functions in the reconcile workstation are provided by a program we developed called Character Verification, a module that allows the reconcile operator to view the bitmapped document images and to verify the text in all the fields, both entered by the edit operator, and that from the automated processes. It is an ActiveX control embedded in the Reconcile software. It is based on two ActiveX controls, the Eastman Kodak Image ActiveX Control and Microsoft's Rich Text Editor ActiveX control.

Character Verification allows the operator to view the image and perform manipulations such as rotation, and zooming in or out. Also a bounding box shows the area on the image that corresponds to the text that the operator is focusing on. The design of this functionality is based on Eastman Kodak Image Control.

Character Verification also allows the operator to edit the field contents. The design of this editor is based on the Microsoft Rich Text Editor and is derived from its heavily used functions such as copy, cut, paste, search and replace. The software can also relate the position of the text characters to the corresponding ones in the image, provide confidence levels, and allow the operator to enter diacritical marks, Greek letters, mathematical signs, change the first character of a selected word to upper case while leaving the rest in lower case, convert case, and complete words in the affiliation field from a partial output.

Character Verification and the Reconcile main program communicate via methods and events. Reconcile sets or gets the methods to instruct Character Verification. In turn, Character Verification fires events back to Reconcile to provide the information.

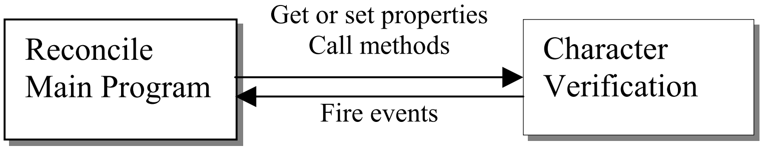


Figure 9.3.3 Reconcile main program communicating with Character Verification.

Character Verification retrieves the OCR output for every article (bibliographic) field, including text contents (character codes), confidence levels and character coordinates, from the MARS database, and the images from a file server. Keyed-in characters are assumed to be at the highest level of confidence and have no coordinates for their location in image.

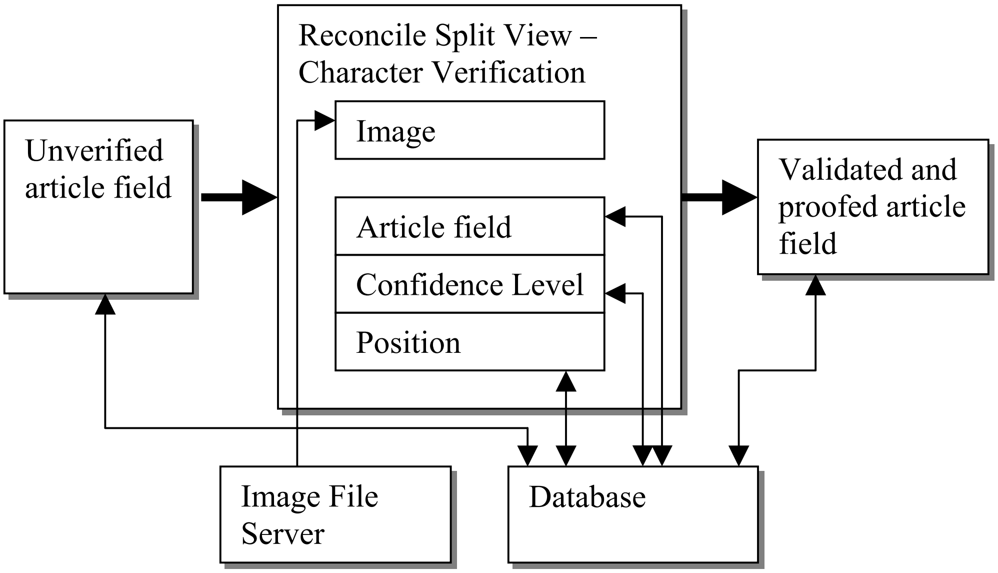
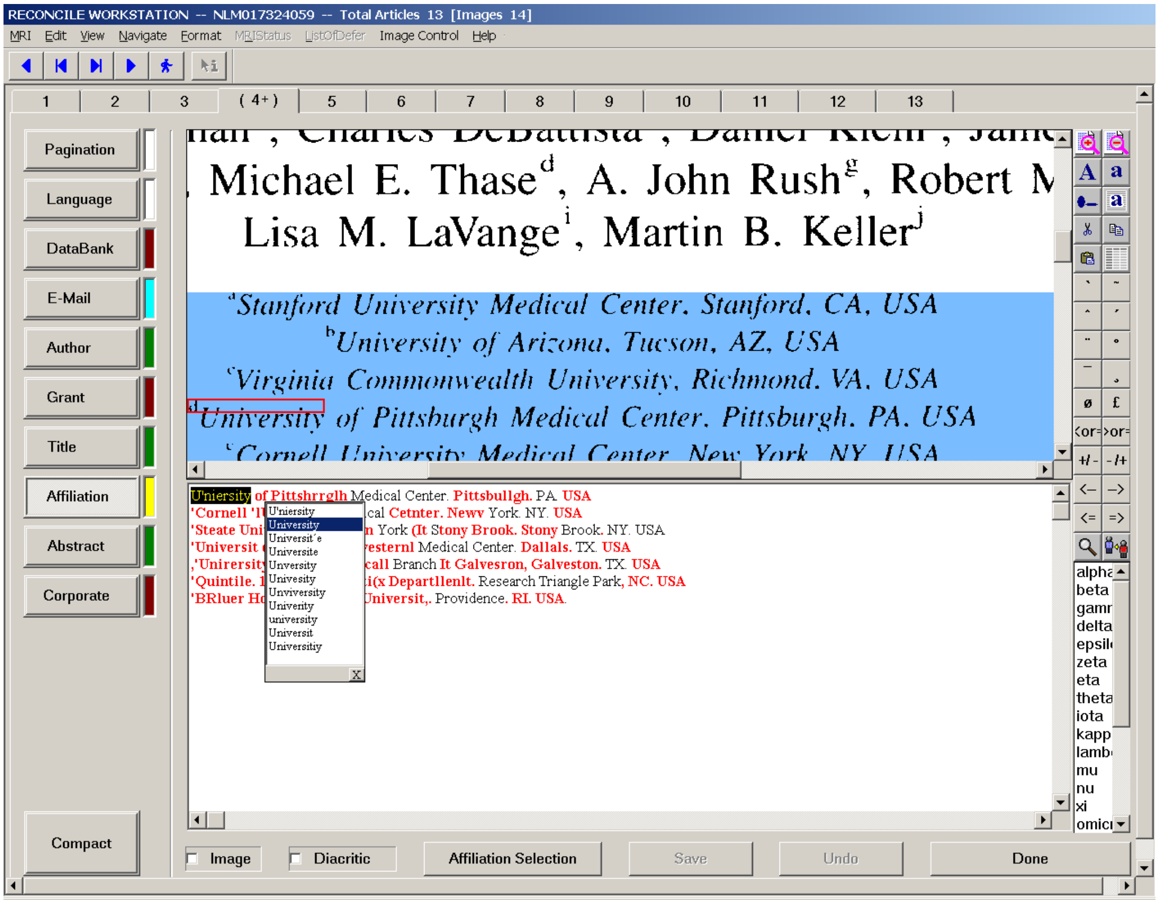


Figure 9.3.4 Character Verification displays label text, confidence and character position.

Character Verification also provides the results of the PatternMatch program to the operator in a word list to choose from. An example is given below showing the operator optional words to select in an affiliation field.



Michael E. Thase[d], A. John Rush[g], Robert M
Lisa M. LaVange[i], Martin B. Keller[j]

[a]Stanford University Medical Center, Stanford, CA, USA
[b]University of Arizona, Tucson, AZ, USA
[c]Virginia Commonwealth University, Richmond, VA, USA
[d]University of Pittsburgh Medical Center, Pittsburgh, PA, USA
[e]Cornell University Medical Center, New York, NY, USA

```
U'niersity of Pittshrrglh Medical Center. Pittsbullgh. PA. USA
'Cornell 'lU          cal Cetnter. Newv York. NY. USA
'Steate Uni           n York (It Stony Brook. Stony Brook. NY. USA
'Universit (          esternl Medical Center. Dallals. TX. USA
,'Unirersity          call Branch It Galvesron, Galveston. TX. USA
'Quintile. 1          i(x Departllenlt. Research Triangle Park, NC. USA
'BRluer Ho            Universit,. Providence. RI. USA.
```

```
U'niersity
University
Universit´e
Universite
Unversity
Univesity
Unviversity
Univerity
university
Universit
Universitiy
```

Figure 9.3.5 Operator can select alternative words in an affiliation field.

10 Production supervision and control tools

In a production system the management and control of the workflow is exercised by the facility supervisors who need to transmit necessary information to the workstation operators. Observing that some of this information flow could be automated, tools were developed to assist in this process. The objective is to meet the supervisors' administrative and management requirements while realizing productivity gains. CheckIn, Admin and CrashPatrol are such tools.

10.1 CheckIn

The first stage in the MARS workflow is CheckIn[44], immediately followed by Scan. CheckIn's role is to collect and organize information necessary to track the physical journal issues from their arrival at the MARS production facility until processing is completed. Prior to the development of this module, the facility supervisor manually researched key information related to the journal issue to be processed, hand wrote the information on a paper form, and attached the form to the issue before sending it on its way to the various operator workstations. CheckIn exploits the fact that most of this information is already in the DCMS (Data Creation Maintenance System) and the MARS databases, and may be automatically extracted from these.

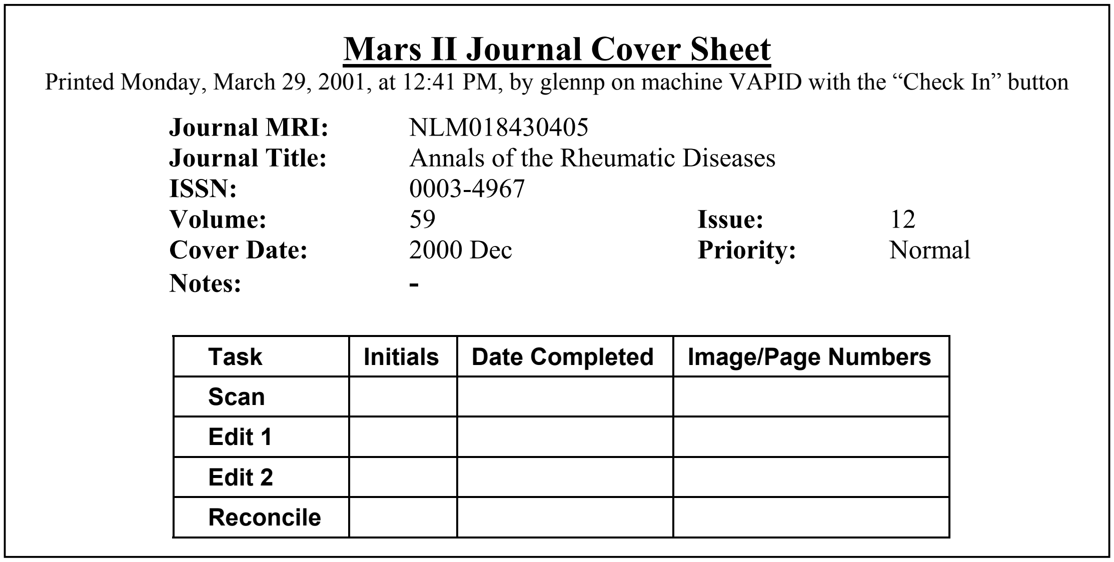
The process is as follows: as a journal issue arrives at the production facility, the supervisor wands the NLM-applied barcode to read the "MRI" number that uniquely identifies the issue. CheckIn sends an on-demand query via network to the DCMS database at NLM, which returns data about the journal title (such as ISSN and journal name) and the particular issue (such as volume number, issue number, and cover date of publication). A concurrent query to the MARS database at the facility retrieves some other items of related information, namely, the journal's priority and notes about how it should be processed.

The retrieved information is used for two purposes:

1) It is printed on a cover page (Figure 10.1.1) that the supervisor paperclips to the journal issue. This information helps keep track of the issue as it moves through the facility and provides guidelines and special instructions for the operators. The cover page is subsequently retained in a manager's notebook as supplemental information.
2) The information, after any adjustment by the supervisor (e.g., change the priority for handling the journal, or to add special instructions for the operators), is stored in the MARS database, initializing the workflow for this issue. The data then flows to the Scan and Edit stations, where the operators do not have to type or pick it manually.

Design and Architecture

The design of CheckIn evolved from the rapid prototyping of several dialog-box-based applications built in VC++/MFC. CheckIn's main dialog window looks like the printed cover page of Figure 10.1.1, minus the grid. CheckIn is a manned application, rather than a daemon, because the "Notes" field, for instance, may need to be edited to pass on supplemental, transient information to the MARS operators.



Mars II Journal Cover Sheet

Printed Monday, March 29, 2001, at 12:41 PM, by glennp on machine VAPID with the "Check In" button

Journal MRI:	NLM018430405		
Journal Title:	Annals of the Rheumatic Diseases		
ISSN:	0003-4967		
Volume:	59	**Issue:**	12
Cover Date:	2000 Dec	**Priority:**	Normal
Notes:	-		

Task	Initials	Date Completed	Image/Page Numbers
Scan			
Edit 1			
Edit 2			
Reconcile			

Figure 10.1.1 **Cover Sheet.** Shown reformatted for conciseness from its actual full-page size. The grid at the bottom is penciled in by each task's operator as the attached physical journal is relinquished. Citational information above the grid, prior to the development of CheckIn, was entered by hand.

Activated by the supervisor, CheckIn passes a journal's MRI as a URL parameter to the DCMS website. Live data is fetched from there in XML format, then parsed using Microsoft's MSXML implementation of the standard XML DOM object[45,47] model. DOM is a good choice for small XML files (as here) since the entire DOM parse tree resides in memory. The data format is defined by the "NLM MEDLINE" DTD[46]. While neither this DTD nor the DCMS web service (i.e., that delivers XML-over-HTTP) were specifically designed for MARS, they offer vital data that MARS needs: ISSN, journal title (in full and short forms), cover date, volume and issue number. CheckIn also retrieves the remaining useful data that is not available with this DTD (e.g., initial/default values for priority, or for in-house "notes" to operators) from the MARS database.

In addition to reading from the MARS database, CheckIn also writes data to it, as purpose (2) suggests. Such database access is through client-side middleware. The original MARS implementation deployed a C++ class library for this: RogueWave's DBTools++. But this has the disadvantage of on-going per-seat costs. The design of CheckIn pioneered our use of Microsoft's ActiveX Data Objects (ADO) instead, a more economical option in the long run.

Besides the addition of role (2) functionality discussed above, CheckIn includes a "Lookup Wizard", to enable the operator to look up a journal from the MARS database by either journal title or ISSN, in case of the following difficulties with retrieving information from DCMS:

1) No response, or an erroneous response, from the DCMS web service;
2) The requested MRI is unknown to DCMS;
3) The returned information is missing an ISSN;

4) The returned ISSN is unknown to MARS.

Case 3 can occur because, in the DCMS database, unlike in MARS, ISSN is neither a key nor a required field. (If a journal really has no ISSN, the facility supervisor enters a placeholder number.) For case 3, CheckIn automatically tries to match the fetched journal title against those in the MARS database. For case 4, the user is asked if such a match attempt by title is desirable.

For all cases, the fallback is Lookup Wizard invocation. The wizard first explains the problem and allows the user to ask for a search by ISSN or title. The second step presents an ordered scrollable list of all journals known to MARS, and an entry field pre-initialized if possible. As the user edits the entry field, automatic prefix matching occurs against the list, so that the field may be completed with minimal typing.

Performance in production

CheckIn's operation has been successful. Each day, the supervisor uses it to check in about 40 journals with at least 600 articles. Prior to CheckIn, the information in the cover pages was hand researched and written, a process that took over an hour. It now takes under 15 minutes. We expect to see additional labor savings upon completion of the rollout of data deposition by CheckIn and its pick up by Scan and Edit.

Next steps

Planned improvements to CheckIn include:

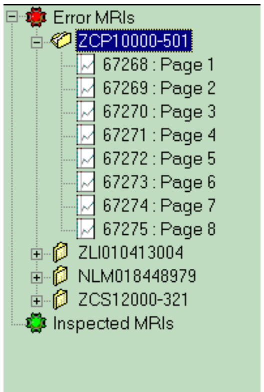
- Enabling the supervisor to add a new journal title and its ISSN to the MARS database at CheckIn, without requiring intervention by the MARS database administrator as at present;
- "Instrumenting" the module so that its performance can be quantitatively assessed;
- Contributing to a projected MARS resource management system, that will replace the cover sheets with a paperless system.

10.2 Admin workstation

When an error occurs, i.e., when a particular journal issue or any of its individual article pages encounter a problem at one of the stages in the workflow, a supervisor ("administrator") has to decide on a corrective step. The Admin workstation software, written in Visual C++ (6.0), allows the supervisor to check on the status of a journal issue in the workflow, and in the event of an error, to request that a process be repeated for a particular journal issue or a page. Every process in the workflow records an error in the ErrorReport table in the MARS database and the process stage is set to 9998 (error stage) in the WorkInProgress (WIP) table. The supervisor uses the Admin module to check the error and decides which MARS process needs to be redone. He/she sends the journal issue back by changing ProcessID and ProcessStage in the WIP table, and switching the integer bits of the PageMasks. The supervisor may also request a particular operator to redo a process by changing the OperatorID.

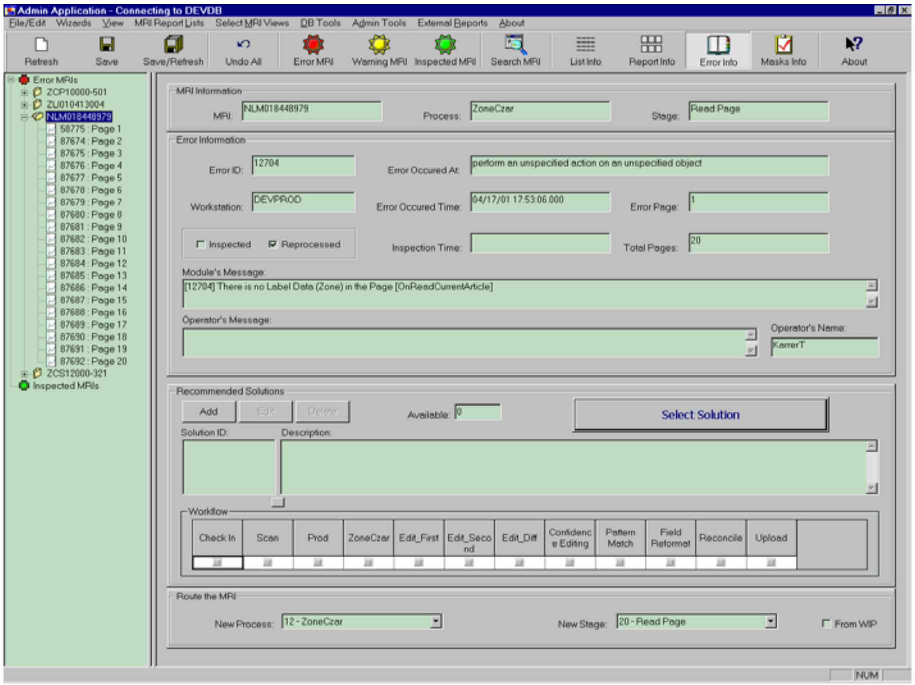
The Admin application is designed to look like Windows Explorer with split windows containing four views, one on the left, and on the right three views which may be switched back and forth for convenience. On the left, the Tree View displays the journal issue affected (MRI number) and its pages. The three views on the right window include: the Error Info View displaying the error information; the List View that opens two other views (List Info and Report Info) to display detailed information pertaining to the journal issue and the individual pages affected; and the Mask View displaying the current page mask settings of the individual pages.

Admin is also designed to adapt to changes in the MARS workflow. Since the workflow may be changed by modifying the ProcessQueue table in the MARS database, Admin builds the data presented on its GUI in line with the new workflow.



- Error MRIs
  - ZCP10000-501
    - 67268 : Page 1
    - 67269 : Page 2
    - 67270 : Page 3
    - 67271 : Page 4
    - 67272 : Page 5
    - 67273 : Page 6
    - 67274 : Page 7
    - 67275 : Page 8
  - ZLI010413004
  - NLM018448979
  - ZCS12000-321
- Inspected MRIs

Figure 10.2.1 Tree View showing the journal issues affected (Error MRIs) and those inspected.



- Error MRIs
  - ZCP10000-501
  - ZLI010413004
  - NLM018448979
    - 58775 : Page 1
    - 87674 : Page 2
    - 87675 : Page 3
    - 87676 : Page 4
    - 87677 : Page 5
    - 87678 : Page 6
    - 87679 : Page 7
    - 87680 : Page 8
    - 87681 : Page 9
    - 87682 : Page 10
    - 87683 : Page 11
    - 87684 : Page 12
    - 87685 : Page 13
    - 87686 : Page 14
    - 87687 : Page 15
    - 87688 : Page 16
    - 87689 : Page 17
    - 87690 : Page 18
    - 87691 : Page 19
    - 87692 : Page 20
  - ZCS12000-321
- Inspected MRIs

MRI Information

MRI: NLM018448979 Process: ZoneCzar Stage: Read Page

Error Information

Error ID: 12704 Error Occured At: perform an unspecified action on an unspecified object

Workstation: DEVPROD Error Occured Time: 04/17/01 17:53:06.000 Error Page: 1

☐ Inspected ☑ Reprocessed Inspection Time: Total Pages: 20

Module's Message: [12704] There is no Label Data (Zone) in the Page [OnReadCurrentArticle]

Operator's Message:

Operator's Name: KarrerT

Recommended Solutions

Available: 0

Solution ID:

Description:

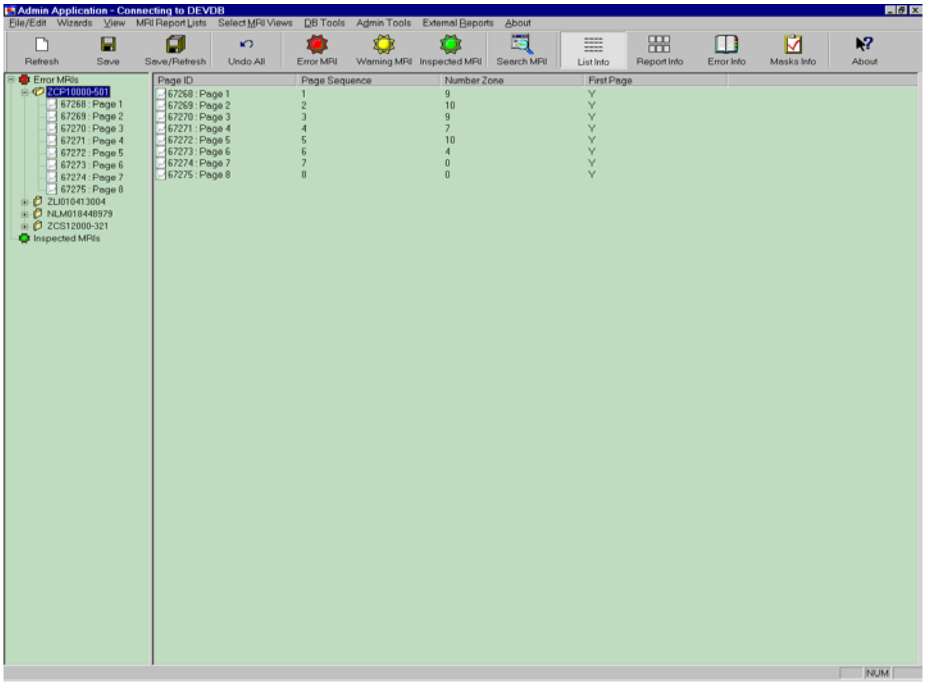
Workflow

Check In	Scan	Prod	ZoneCzar	Edit_First	Edit_Second	Edit_Diff	Confidence Editing	Pattern Match	Field Reformat	Reconcile	Upload

Route the MRI

New Process: 12 - ZoneCzar New Stage: 20 - Read Page ☐ From WIP

Figure 10.2.2 Error Info View on the right displays the information on journal issues or individual pages



Page ID	Page Sequence	Number Zone	First Page
67268 : Page 1	1	9	Y
67269 : Page 2	2	10	Y
67270 : Page 3	3	9	Y
67271 : Page 4	4	7	Y
67272 : Page 5	5	10	Y
67273 : Page 6	6	4	Y
67274 : Page 7	7	0	Y
67275 : Page 8	8	0	Y

Figure 10.2.3 List Info view

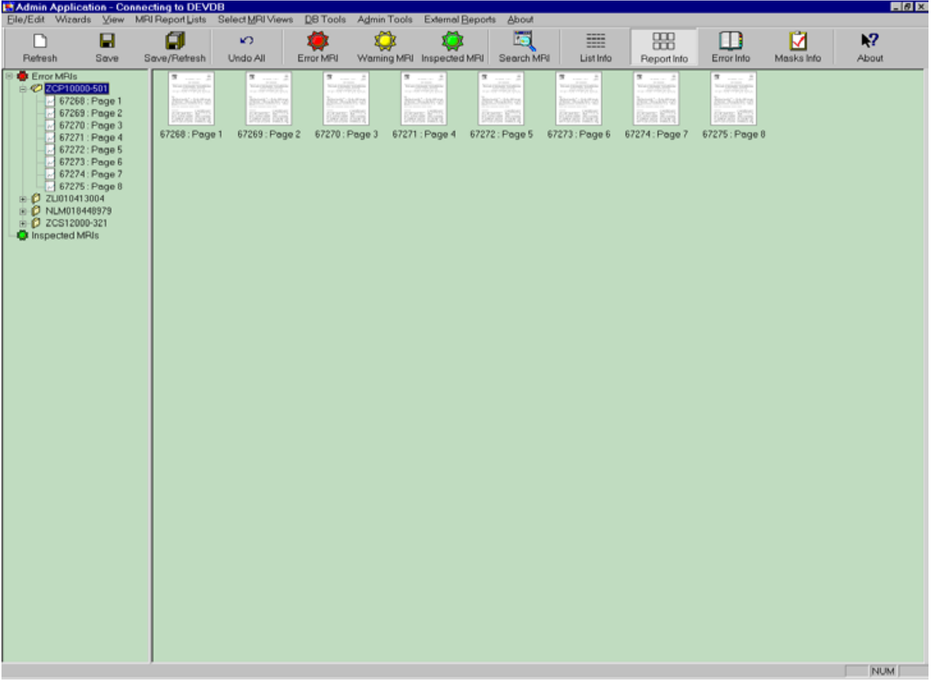
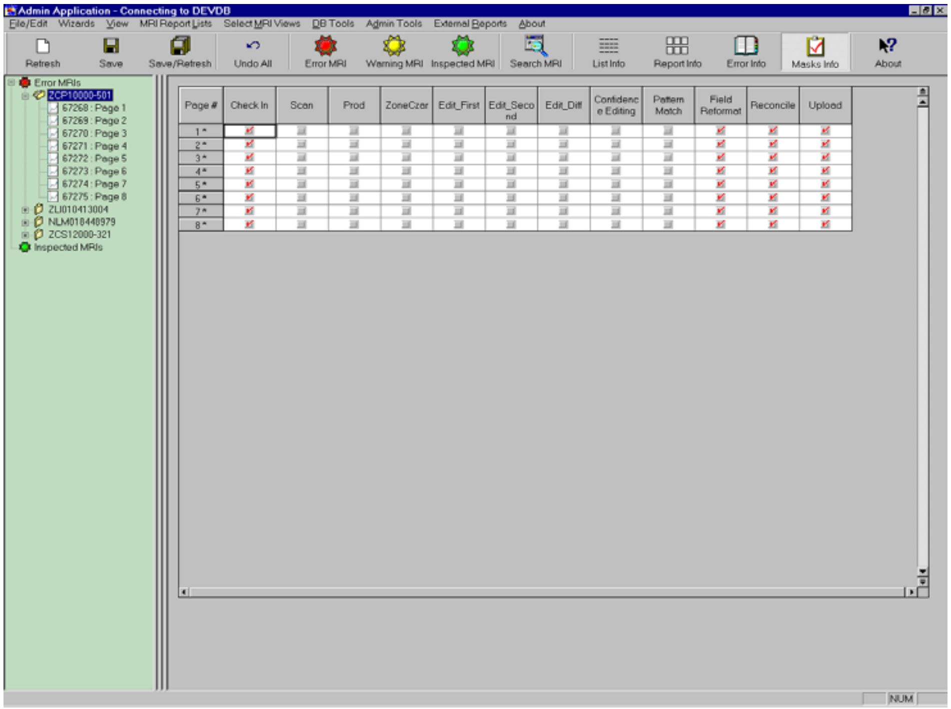


Figure 10.2.4 Report Info view



Page #	Check In	Scan	Prod	ZoneCzar	Edit_First	Edit_Second	Edit_Diff	Confidence Editing	Pattern Match	Field Reformat	Reconcile	Upload
1 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
2 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
3 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
4 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
5 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
6 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
7 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑
8 *	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑

Figure 10.2.5 Mask View displays Page Mask. The masks can be set up so the 32-bit integer is set for reprocessing

The functionality of Admin is as follows. The supervisor selects a particular journal issue (MRI) on the tree view, and information related to that MRI is displayed: the PageID, PageSequence, NumberZone, FirstPage are displayed on the first view, LabelID, LabelField, LabelType, Confidence (if page is selected on the tree view.)

The Information View displays MRI, current process, current stage, module message, user message, error information, and error solution. The new process and new stage are displayed in a combo box to assist the supervisor to choose a new process for reprocessing the journal issue.

The supervisor can use the Mask View to set up the PageMask by clicking on the **Select Solution** button after he/she sets the new process mask or a new process id or process stage. A dialog box appears to allow the supervisor to choose the pages that need reprocessing. The Mask View shows which process(es) failed and requires to be redone. The information pertaining to the new choices will be saved to the database for reprocessing.

Admin is designed so that a separate thread checks the ErrorReport table every 3 minutes for a new record. If one exists, it will be added to the tree automatically. Also, a blinking red light indicates there is at least one error to be handled.

Other actions the supervisor can take include:

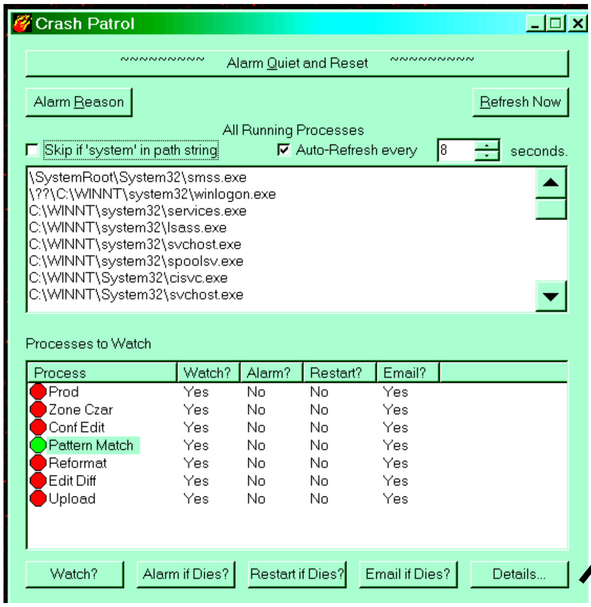
- Double clicking on the MRI in the tree view activates an embedded Label Browser that can be used for browsing the data.
- Double clicking on Page in the tree view results in displaying the image of that page on a modeless dialog box, so the supervisor may move from page to page to load the corresponding image.
- Double clicking on the Label (bibliographic field) in the list view results in a display of the text in that label in a modeless dialog box.
- Using the Transfer Data/Object dialog box, the supervisor can transfer data from one server to another.
- Using the System Type Check dialog box, the supervisor can check whether the journal issue is compliant (amenable to automated processing) or not.

A future functionality will be an audit trail to keep track of all the actions of supervisors when using the Admin system.

10.3 Crash Patrol

Since any downtime in a production system is a serious matter and needs to be minimized, it is important to have an automated alert in case of failures. The purpose of the Crash Patrol subsystem, implemented as a multi-threaded Windows C++ application, is to alert the supervisors and technical support personnel automatically in case a MARS daemon fails. This module is designed to give a quick warning of the crash of an executable program, and complements other database-centered techniques that also detect deadlock or inactivity, though more slowly. When such a crash occurs, Crash Patrol sounds an audible alarm through the facility's speaker system. The alarm volume may be customized for each process that is monitored. Remote notification by email is also possible. To use this feature, the current implementation requires installation of the Windows NT/2000 SMTP Server, though for the future, a version that handles SMTP more independently might better allay security concerns.

Crash Patrol also has the capability to restart crashed programs, although most of the MARS daemons have been programmed to restart without manual intervention. If restart is requested, Crash Patrol will sound an alarm only if the restart fails.



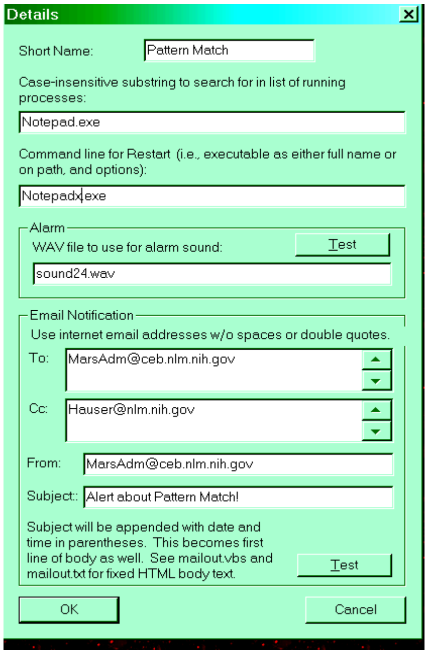
All Running Processes

☐ Skip if 'system' in path string ☑ Auto-Refresh every 8 seconds.

```
\SystemRoot\System32\smss.exe
\??\C:\WINNT\system32\winlogon.exe
C:\WINNT\system32\services.exe
C:\WINNT\system32\lsass.exe
C:\WINNT\system32\svchost.exe
C:\WINNT\system32\spoolsv.exe
C:\WINNT\System32\cisvc.exe
C:\WINNT\System32\svchost.exe
```

Processes to Watch

Process	Watch?	Alarm?	Restart?	Email?
🔴 Prod	Yes	No	No	Yes
🔴 Zone Czar	Yes	No	No	Yes
🔴 Conf Edit	Yes	No	No	Yes
🟢 Pattern Match	Yes	No	No	Yes
🔴 Reformat	Yes	No	No	Yes
🔴 Edit Diff	Yes	No	No	Yes
🔴 Upload	Yes	No	No	Yes



Short Name: Pattern Match

Case-insensitive substring to search for in list of running processes: Notepad.exe

Command line for Restart (i.e., executable as either full name or on path, and options): Notepadxexe

Alarm

WAV file to use for alarm sound: sound24.wav

Email Notification

Use internet email addresses w/o spaces or double quotes.

To: MarsAdm@ceb.nlm.nih.gov

Cc: Hauser@nlm.nih.gov

From: MarsAdm@ceb.nlm.nih.gov

Subject:: Alert about Pattern Match!

Subject will be appended with date and time in parentheses. This becomes first line of body as well. See mailout.vbs and mailout.txt for fixed HTML body text.

Left window. The main window of this dialog application has, at the top, a snooze bar to turn off the alarm sound. A button (Alarm Reason) offers textual information about why the alarm sounded, although with experience this may be quickly elicited from the lower "Processes to Watch" window. The top list pane shows all processes running on the machine, similar to the Task Manager in Windows NT/2000; system processes can be screened out. This pane is initialized at startup and then refreshed either manually or at a desired frequency. Substring search within this list is the detection mechanism. The lower pane shows seven MARS daemons. For each, a traffic light is either green if it is running, red if it has stopped, or gray if it does not matter (indicated by toggling "Watch?" to "No"). Toggling of the Yes/No columns is done with buttons at the bottom.

Right window. This window appears on pressing the "Details" button in the main window. The example shown here is while exercising the alarm not with the real Pattern Match, but with Notepad. The restart command line shown has bad-text "notepadx.exe" (instead of, say, "notepad.exe") to test failure to successfully restart the process, leading to alarm triggering.

11 System performance evaluation

Assessing the performance of the MARS system is an important goal, not only to evaluate the efficiency of its constituent modules, but also to locate potential bottlenecks. In addition, since we seek the best way to create MEDLINE bibliographic records, it is important to compare the productivity (e.g., labor hours per unit record) of the MARS systems, both versions 1 and 2, against each other, as well as with that of the manual keyboarding operation done under contract. Key questions posed as a starting point for performance evaluation are listed as follows:

1. How long does it take for a bibliographic record to be completed?
2. What is the time taken by each manual and automatic process for one record?
3. What is the time taken by each manual and automatic process for one day's workload?
4. What is the time taken to enter each field in Edit?
5. What is the error rate of the zoning, labeling and reformat modules?
6. What is the utilization rate of MARS-2 server processes and workstations?
7. How long does data wait to begin processing by each of the daemon process? I.e., how long is it in a queue waiting to begin work?
8. How often is a citation re-processed? What are the most common reasons?
9. What is the overall cost (in labor-hours) for the MARS-2 operation as compared to MARS-1 and the keyboard operation?

These questions are addressed quantitatively by instrumenting the system and analyzing the data recorded, these mainly being event counts and time data. Instrumentation is implemented by two C++ classes written to record such data: ProcessTime which records times and PerformanceData that records statistics generated in a MARS process.

11.1 Process performance analysis

The instrumentation data yields information on the processes, both automatic and manual, at different levels of granularity. Figure 11.1 shows the average time taken by each process to complete its task for one bibliographic record (citation) in July 2001. Predictably, the manual processes of scanning, editing and reconciling take much longer than the automated ones. An explanation of the terminology: Edit_First and Edit_Second stand for the first and second Edit operator; Prod is the inhouse-developed daemon that controls the OCR system, hence equivalent to the OCR action; ZoneCzar combines the actions of automated zoning and labeling.

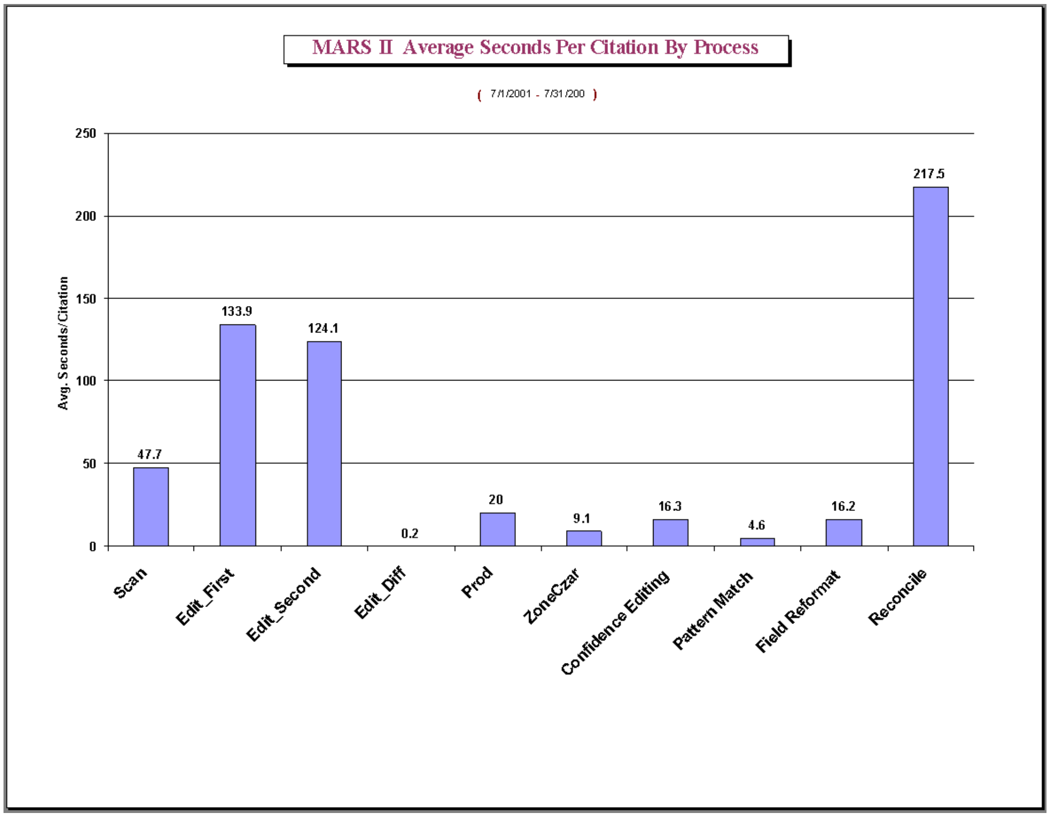


Figure 11.1

Instrumentation data for a breakdown of some of these processes into their constituents appear in Figures 11.2 and 11.3. In Figure 11.2 we find the actual process of scanning a page (“append”) to take a relatively short time, but inserting a missing page after the fact and the entry of a new journal issue (“New MRI”) to take much longer. This latter task, found time consuming, was the rationale behind the development of the new CheckIn module, which eliminates this function in the scanning operation.

The actual workload for these time consuming processes is not high, because they do not occur frequently. For example, as shown in Figure 11.3, the actual burden of inserting pages is very low, since this operation is performed rarely.

Figure 11.4 shows the average time taken for the Edit operator to enter the fields not automatically extracted. Only the data for the first Edit operator is shown, since the data for the second operator is approximately the same. In this figure, we show entries for those fields that are automatically extracted in compliant journals, because we are accommodating non-compliant ones also. Furthermore, even for compliant journals, there are pages that are not processed by the automatic modules (e.g., letters to the editor, editorials) requiring the Edit operator to key in the relevant data. The figure, however, indicates opportunities for further automation.

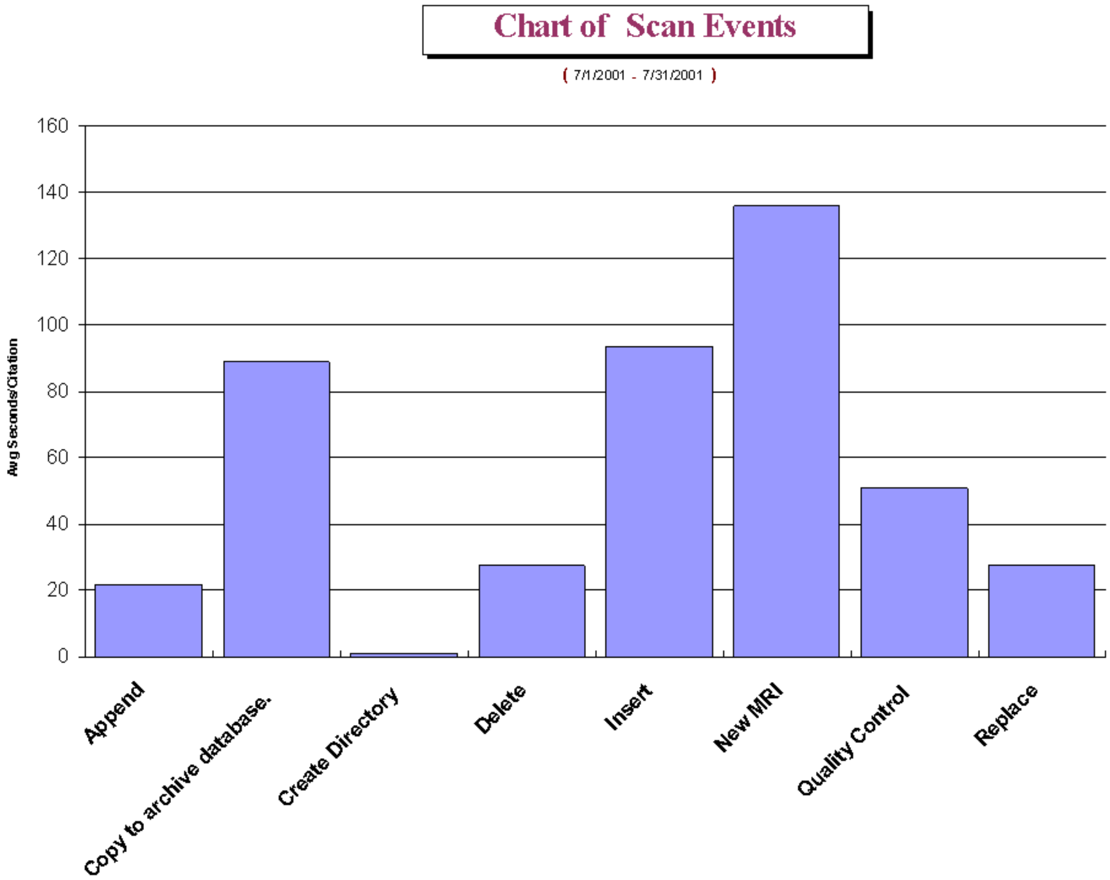


Chart of Scan Events
(7/1/2001 - 7/31/2001)
Avg Seconds/Citation
160
140
120
100
80
60
40
20
0
Append
Copy to archive database.
Create Directory
Delete
Insert
New MRI
Quality Control
Replace

Figure 11.2

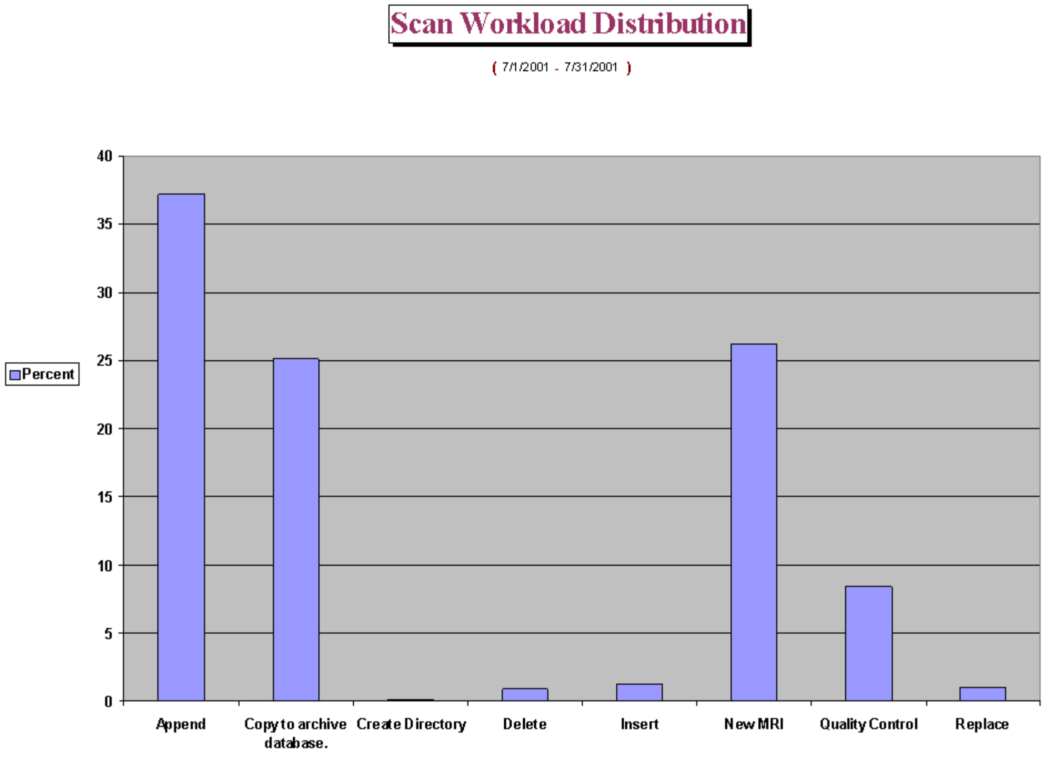


Figure 11.3

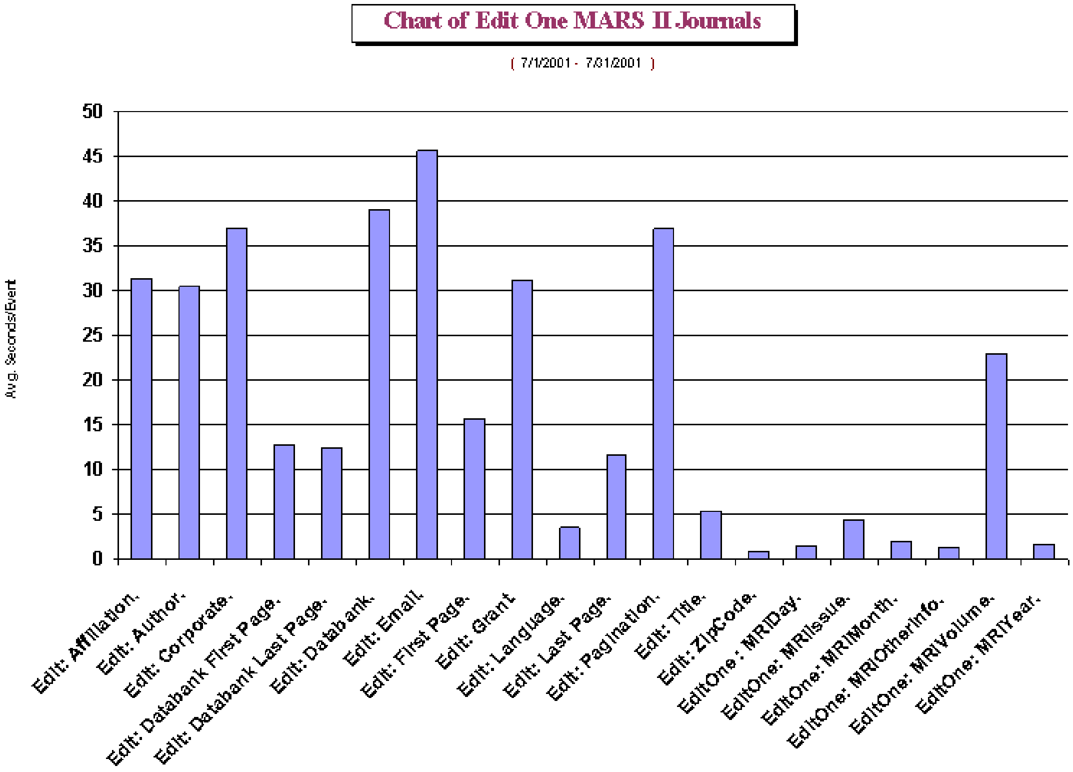
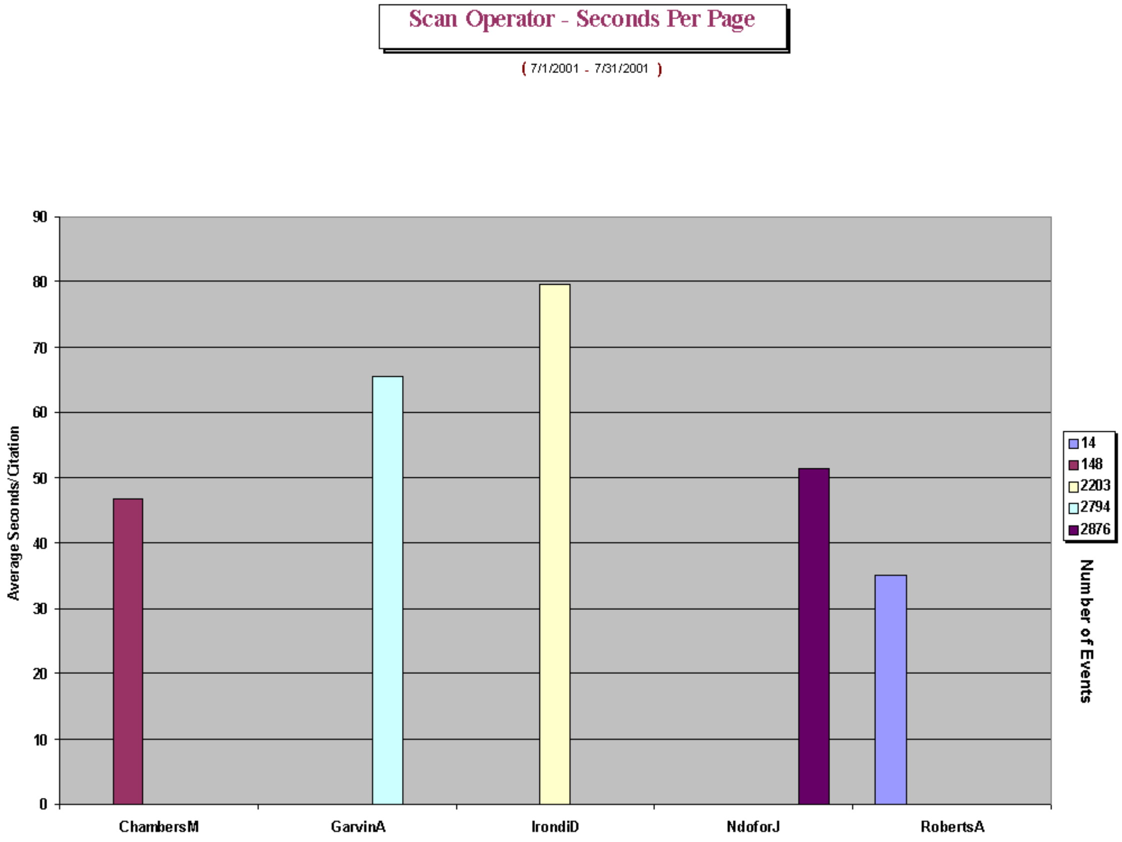


Figure 11.4

Since we keep track of operator names in the database, we also offer the supervisor the option of comparing their relative effectiveness, as shown in Figure 11.5 for scanning and Figure 11.6 for editing.

Figure 11.5



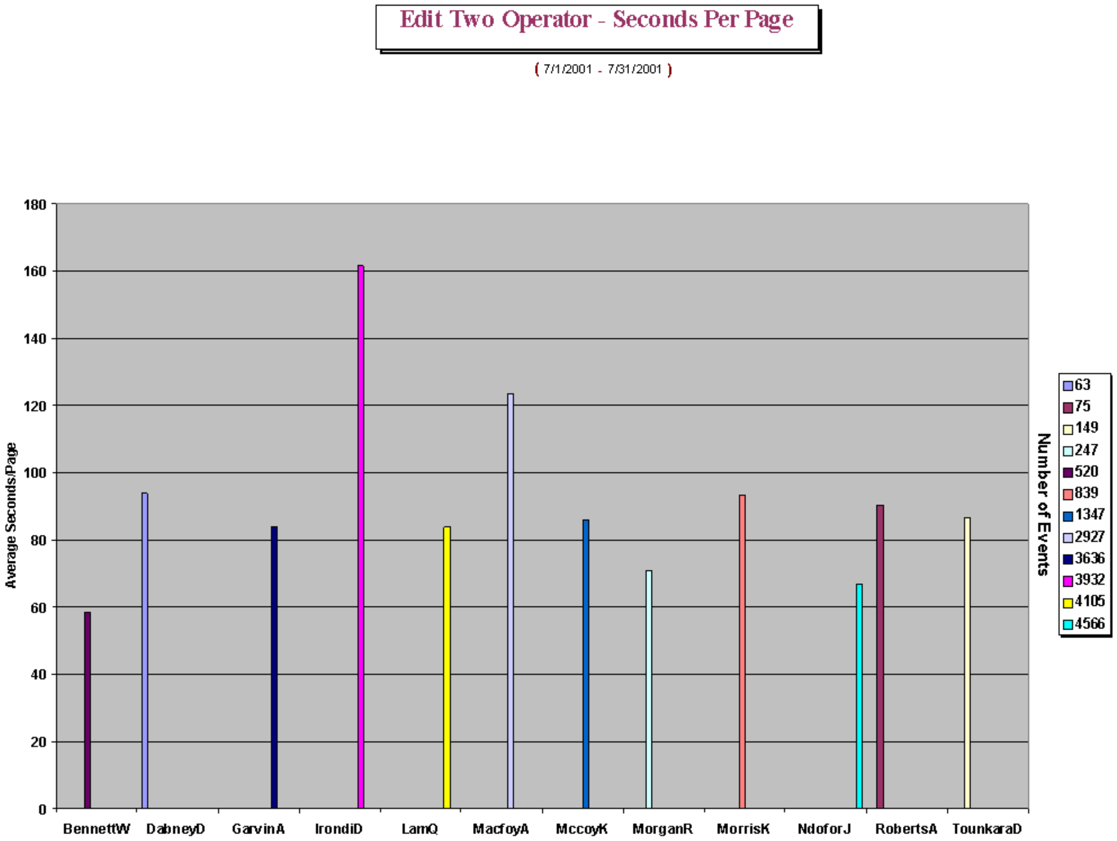


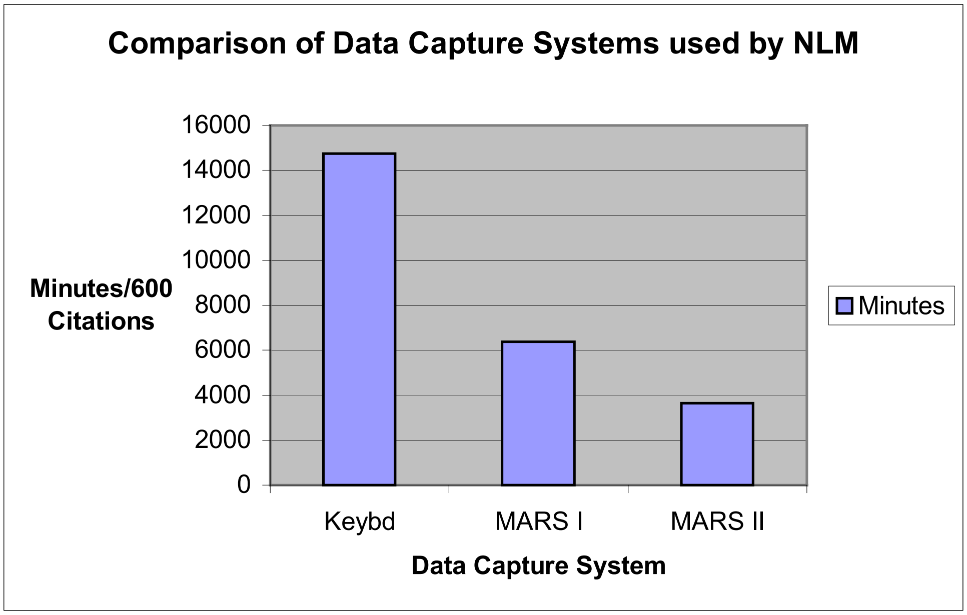
Figure 11.6

11.2 Comparison of the three data entry systems

Here we compare the two systems, MARS-1 and MARS-2, and the manual keyboarding operation on the basis of a workload of 600 completed bibliographic records per day, the average workday load for all of these approaches. The table lists the average number of seconds per page for each system and the number of minutes per 600 records, and shown in a chart in Figure 11.7. It can be seen that MARS-2, by eliminating many of the manual functions in MARS-1, is a considerable improvement, and that both are far more efficient than the manual keyboarding operation. To produce 600 records, MARS-2 requires 61 hours of labor per day, while the keyboarding requires 246 hours. In comparison with the keyboarding operation, MARS-2 therefore saves 185 direct labor-hours per day or 51,800 labor-hours per year (based on a year of 280 work days).

	KeyBd	**MARS I**	**MARS II**	**Keybd**	**MARS I**	**MARS II**
Category	***Sec/Page***	***Sec/Page***	***Sec/Page***	**Min/600**	**Min/600**	**Min/600**
Scan	NA	71	30	NA	706	300
Edit	NA	178	133	NA	1784	1330
Reconcile	NA	388	202	NA	3885	2020
Total	1475	637	365	14750	6374	3650

Figure 11.7


Comparison of Data Capture Systems used by NLM
Minutes/600 Citations
16000
14000
12000
10000
8000
6000
4000
2000
0
Keybd
MARS I
MARS II
Data Capture System
🟦 Minutes

12 Next tasks

In individual sections we have outlined tasks that will improve the automatic and manual processes in MARS. In addition, we seek to initiate new projects that go beyond such incremental improvements. These include: creating a ground truth database for research in document image analysis and understanding; a system to extract bibliographic data automatically from online journals; and an alternative method that could improve the productivity of the reconcile (final verification) stage in the system.

12.1 Ground truth data: PathFinder

12.1.1 Introduction and objective

Successful document image analysis is greatly dependent on ground truth data for the design, training and testing of algorithms for data identification and extraction. However, ground truth datasets and their associated analysis and visualization tools are usually created to analyze problems in specific datatypes: skewed document images (Okun et al.)[48], handwritten documents (Cha and Srihari)[49], video sequences (Doermann and Mihalcik)[50], statistical data (Swayne et al.)[51], and speech signals (Barras et al.)[52]. Moreover, apart from the domain-specific nature of these datasets and tools, they are usually limited as to operating platforms and data formats, as described in an excellent taxonomy on this subject by Kanungo et al.[53]

To our knowledge no ground truth dataset exists that represents the corpus of biomedical journals, and none with the goal of extracting the text representing the bibliographic fields descriptive of the articles within these journals.

In meeting this challenge, the main objective of the PathFinder (***P**ublic **A**rchive **T**o **H**elp **Fi**nd **N**ew **D**esigns for **E**xpert **R**atiocination*) project is to exploit the vast amounts of document images and OCR-converted and operator-verified data, already collected in the MARS project, to aid in the development of innovative and efficient algorithms for automatic zoning, labeling and reformatting by the computer science and medical informatics communities. This is to be done by developing a ground truth database accessible via the Web. This ground truth data will include page, zone, line, word, and character level information. In addition to providing a public site for researchers worldwide to develop and test their algorithms, this system will enable them to graphically visualize the ground truth data and employ an automated analysis assistant. Code-named Rover (*g**RO**undtruth **V**s. **E**ngineered **R**esults*), this automated analysis assistant will compare the results of a researcher's program to the ground truth data. The ground truth and results data will be in XML and MARS Rover will be written in Java. The overall website development will use MacroMedia Dreamweaver UltradDev 4 to provide a rich interface and extensive database connectivity.

12.1.2 Design considerations

Page layout. Identifying geometric features to design rule-based algorithms for automated data extraction is a non-trivial task since (as discussed earlier in this report) there is a variety of layout geometries in the 4,300 journal titles indexed in MEDLINE, though most follow the reading order paradigm of article title-author names-author affiliation-abstract. Ground truth data must include samples of all major layout types classified as follows:

- Series 10000 – The abstract is in a single column and covers the width of the page. There are 13 sub-types in this series. The variations cover the location of the affiliation field.
- Series 12000 – The abstract is in a single column in a double column layout. It is always in the left hand column followed by the body of the article. There are 13 sub-types in this series, the variations again covering the location of the affiliation.
- Series 12200 – The abstract appears in two adjacent columns, and the affiliation occupies the full width of a single column or might span a double column. There are 13 sub-types in this series involving the location of the affiliation.
- Series 99999 – This series includes journals that do not follow the usual reading order paradigm. There are over 1,000 journal titles that exhibit these non-standard formats, and offers challenges to automation.

Data format. The data format of choice is XML for images, OCR-converted data and operator-verified data since XML excels in adaptation (to accommodate changes in data), maintenance, linking (from one piece of data to another), simplicity, and portability over networks, operating systems and development environments. Moreover, XML is growing in popularity and its strengths have been proven in existing MARS modules such as CheckIn and Upload. One of the first tasks in the PathFinder project will be to select and convert the MARS ground truth data into XML.

Modifying existing data. While rich in the information it provides, the existing data has certain deficiencies. For example, although OCR output characters in error are corrected, their attributes (e.g., *italics* or **bold**) are not. But these attributes can serve as features to create algorithmic rules to correctly identify zones or labels. Therefore, some effort will be made to correct these before including them in the ground truth dataset.

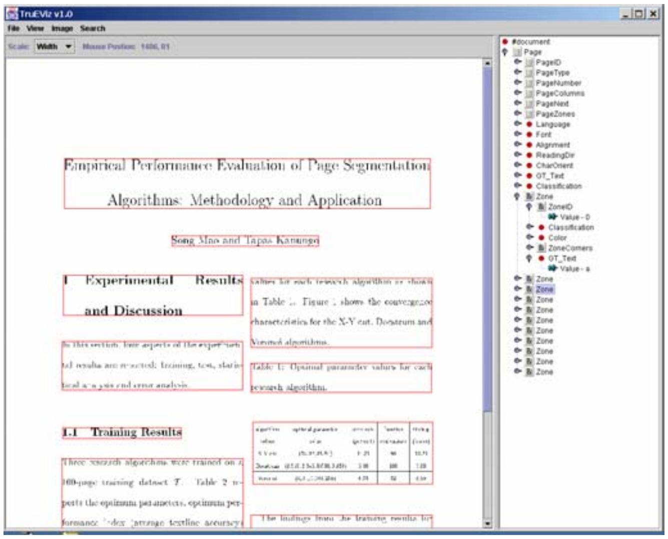
12.1.3 Rover: a visualization and analysis tool

Once the data is available in XML format along with the original TIFF image, researchers need the functionality to:

- Load a TIFF image and the corresponding XML file into an application, and display the XML data, where appropriate, overlaid on the image.
- Modify and add new ground truth data.
- Compare the results of new algorithms against the ground truth data.

A survey of visualization tools[53] identified TrueViz as a suitable platform for the design of Rover, since it provides the first two features noted above. TrueViz is a public domain tool developed at the University of Maryland for visualizing, creating and editing ground truth and metadata. It is implemented in Java and works on Windows and Unix platforms (specifically, it has been successfully tested in the Windows 2000 and Sun Solaris 2.6 environments). It reads and stores ground truth and metadata in XML format, and reads the corresponding TIFF images. It has the ability to allow the user to inspect ground truth data at many levels, viz., at the page, zone, line, word, and character levels, and provides pertinent information at each level. For example, at the character level, such information includes the character code, font type and style, and bounding box (x1,y1,x2,y2) coordinates.

To serve as an effective analysis assistant, Rover will extend TrueViz to provide researchers the capability to *compare* their XML data to MARS ground truth XML data graphically, and thereby serve as a powerful tool in helping them iteratively refine their algorithms. In addition, it will provide numerous statistics and visual presentations on specified areas. For example, the user would be able to use Rover to compare all characters in the dataset that are **bold,** and to enumerate errors. Rover would visually locate the mistakes and report statistics on the query. In this example it would report the number of bold characters, the number detected correctly, and the number detected incorrectly, both as absolute numbers and as percentages. Rover would also export this information to a database or spreadsheet for further analysis.



Empirical Performance Evaluation of Page Segmentation Algorithms: Methodology and Application

Song Mao and Tapas Kanungo

1 Experimental Results and Discussion

[illegible] in Table 1. Figure 1 shows the convergence characteristics for the X-Y cut, Docstrum and Voronoi algorithms.

Table 1: Optimal parameter values for each research algorithm.

1.1 Training Results

Figure 12.1.1 Rover screen snapshot of TIFF page and zone segments.

12.1.4 Ground truth distribution

A website will provide access to the ground truth data, though it will go beyond serving as a simple data repository. Our objective is to encourage researchers to share and exchange ideas, as well as provide feedback to our development team for new desirable features. Figure 12.1.2 shows the organization of the website and how the elements interconnect. This section presents an overview of the website layout and functionality.

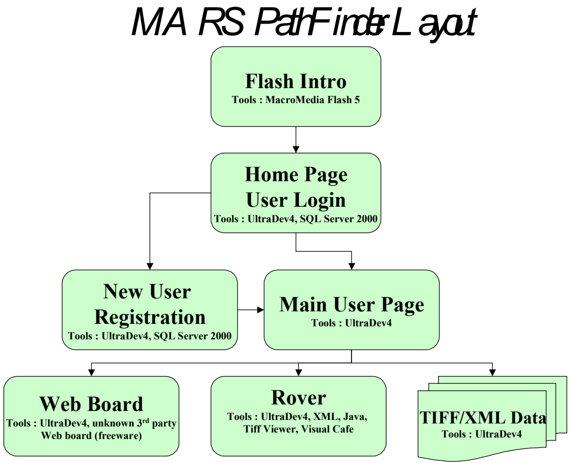

MA RS PathFinder Layout
Flash Intro
Tools : MacroMedia Flash 5
Home Page
User Login
Tools : UltraDev4, SQL Server 2000
New User
Registration
Tools : UltraDev4, SQL Server 2000
Main User Page
Tools : UltraDev4
Web Board
Tools : UltraDev4, unknown 3rd party Web board (freeware)
Rover
Tools : UltraDev4, XML, Java, Tiff Viewer, Visual Cafe
TIFF/XML Data
Tools : UltraDev4

Figure 12.1.2 – Main sections of Website and tools

Product introduction. A Macromedia Quick Flash "movie" is displayed to the users, though this may be bypassed. We have already developed this movie and a demonstration of it is available at www.geocities.com/egalthan/Movie1.html. The metaphor used is a Sherlock Holmes style magnifying glass moving over a rare biomedical document. Figure 12.1.3 is a screen snapshot of the final frame of the movie.

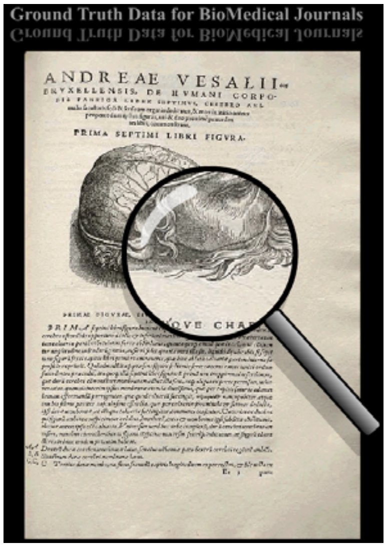

Ground Truth Data for BioMedical Journals
ANDREAE VESALII
PRIMA SEPTIMI LIBRI FIGVRA.

Figure 12.1.3 The final frame of the opening Macromedia Flash Movie.

User login. On the main page of the website, users are asked for a name and password. This data will be stored in a SQL Server 2000 database along with the website data. We intend to use Macromedia Ultradev as the design tool, because it readily allows for password protection and database connectivity. While we do not anticipate restricting anyone from accessing this site, we intend to require all users to register. When registered users log in, they will be taken to the main user page. Unregistered users will be presented with a registration page before being allowed access.

New user registration. The system will require first time users to enter a few important items of information that will enable us to provide security to the system as well as the ability to contact users with important new data or features added to the website in the future. The registration page will be developed using Ultradev which allows the developer a graphical design environment and provides rapid page development.

Data access. Once logged in or registered, the user would have access to all the ground truth data and tools. Since the data collection will be quite large, the system will allow users to download the entire data set or any subset they choose. For example, some users may only be interested in certain types of journal layouts, such as double column abstracts. The ground truth data will be organized in a hierarchical fashion as shown below.

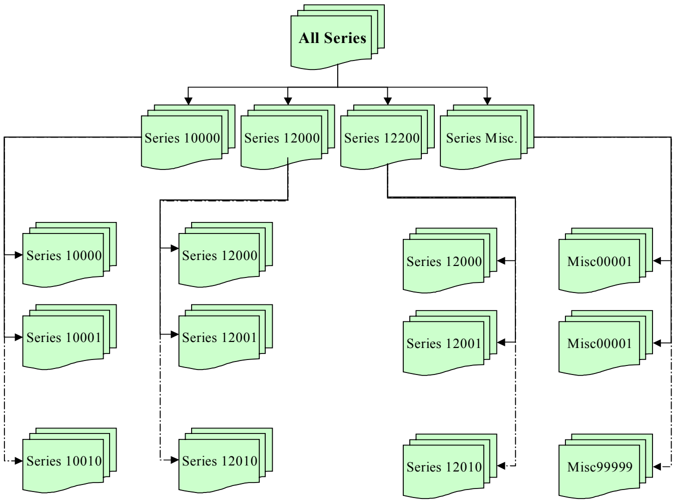


Figure 12.1.4 Hierarchical tree structure for storage and retrieval of ground truth data

Data analysis. The users will be provided a link to launch Rover. While the initial version of this tool will possess the current functionality of TrueViz, the complete analysis support discussed earlier will be designed in a later phase. At present, TrueViz may be used to view ground truth data and research data (e.g., results of algorithmic processing) following the XML defined structure. Rover will provide the users the ability to automatically compare ground truth data against research data, and perform statistical analysis tasks.

Bulletin board. This section of the website will be available to the user community to report bugs, and use as a forum for discussions related to algorithmic development.

Upload and download area. This section of the website is for the users to upload and download files, such as technical papers written, algorithm source code, and new ground truth data.

12.2 Data extraction from online journals

12.1 Background

With the rapid expansion and utilization of the Internet and Web technologies, there is an increasing number of online biomedical journals, some of which are being indexed in MEDLINE. Online journals pose new challenges in the areas of automated document analysis and content extraction, database citation records creation, data mining and related applications. New techniques are needed to capture, classify, analyze, extract, modify, and reformat Web-based document information for computer storage, access, and processing. We propose the design and development of an automated system, temporarily code-named WebMARS for Web-Based Medical Article Records System, to create citation records for MEDLINE. This system[54] will download and classify document articles on the Web, parse and label contents of each article, extract and reformat the bibliographic fields from the article, display the entire citation to operators for reconciling (validation), and upload the citation records to the MEDLINE database.

12.2 Project objectives

The objectives are to design, develop, evaluate and implement a production system that will automate the extraction of bibliographic data from online journals on the Web for the MEDLINE database.

12.3 Project significance

With the increasing role of online journals in biomedical publishing, it is imperative that they be processed automatically to extract bibliographic data with a minimum of human labor. Two obvious advantages of such a Web-based document analysis and content extraction approach compared to either manual keyboard entry or the scanning/OCR approach are saving labor costs and improving performance (speed and accuracy).

12.4 System overview

The proposed WebMARS system would consist of seven servers and two types of operator workstations: *download and classification*, and *reconciling.* A high-level diagram of the system is shown in Figure 12.2.1. All workstations and servers are networked via a LAN and communicate through the WebMARS database server.

Briefly, the WebMARS system will work as follows. An operator downloads an online journal from a publisher's website, and the PDF or HTML files of the articles are sent to the file server. The *PDF to HTML files conversion server* performs text conversion. The *article zones creation and labeling server* extracts text zones from the HTML files, and labels these zones as belonging to "title", "author", "affiliation", "abstract", “pagination”, “databank accession number”, “grant number”, etc. The *article zone contents modification server* extracts, modifies, and reformats the text according to MEDLINE database conventions. The *MEDLINE article zones creation and Web-based reconciling workstation* selects and combines the appropriate portions of zone contents to create records that are viewed by operators for validation and reconciling. The reconciling workstation operator then checks, proofreads, and corrects any remaining problems in the records including character and symbol validation, citation reformatting, field label selection and confirmation. The *SGML-based MEDLINE citation records creation server* generates text zones directly from journal publisher SGML information, if available, that can be used to further improve labeling process. Finally, the *MEDLINE citation records uploading server* uploads the completed citation records to the NLM's DCMS database for later indexing.

12.5 System servers and workstations

Each subsystem of the WebMARS system is described below.

Web files collection and classification workstation. This workstation downloads and classifies Web-based files as abstract, full text, PDF or image files[55]. It consists of two processes: *downloading* and *classification*. The first is based on *WinInet* software tool and a combination of the *Breadth First Search* algorithm and the *Constraint Satisfaction* method to traverse the Web page's links, recognize and generate the successors of the downloading pages. The second relies on the contents of the hyperlinks (name and address) to classify the files.

The downloading process shown in Figure 12.2.2 selects the Web address of the start page, and sends a request to the Web server to download that page. During download, the links of the current downloading page are classified, and the children are generated and added to the end of the open list (consisting of addresses of pages waiting to be downloaded). Then the successfully downloaded page is classified and saved to a directory in the file server corresponding to its type, and the address of the start page is added as a node to the closed list (consisting of addresses of pages that are successfully downloaded). The open list now contains all of the successor links of the downloaded page that are waiting to be downloaded. The downloading process continues selecting the first node of the open list to download. If the download is successful, the node is removed from the open list and added to the closed list. Otherwise, the node is_put into the revisited list (consisting of addresses that failed during the downloading process, and are to be revisited later). The children of the successfully downloaded page are determined by the classification process and added to the end of the open list. The entire process is repeated until the open list is empty.

The classification process validates the hyperlinks to make sure that they have satisfied the predefined constraints for saving the downloaded Web page and generating its children. Since the contents of the hyperlinks (name and address) consist of useful information about file types, these can be used to determine the appropriate storage for saving the documents and the satisfied child links of the currently downloading page. The downloaded Web pages are saved in different directories in the file server corresponding to their types. A hyperlink, which satisfies the constraints to be a potential successor of the current downloading web page, is generated as a child node and added to the open list. Otherwise, it would be eliminated.

PDF to HTML files conversion server. Occasionally, publishers offer certain Web journal articles as PDF files rather than as HTML files. Since WebMARS requires the HTML file format, this conversion server subsystem is used to convert downloaded PDF files into HTML files with available commercial software.

Article zones creation and labeling server. This server subsystem first parses and creates text zones for each journal article based on its HTML file, and then labels these text zones. The subsystem consists of three modules: the *automated zoning module*, the *automated labeling module*, and the *automated updating module*.

The automated zoning module relies on HTML tags to parse and then to create text zones for each article of a journal issue.

The automated labeling module labels these text zones using a combination of statistics and fuzzy rule-based technology. Statistics are used to generate membership functions for the fuzzy rules-based method. Features and fuzzy rules derived for the automated labeling module are based on an analysis of the HTML layouts of journals. Both geometric and non-geometric features are considered here. Geometric features of a zone are based on location and order of appearance. Non-geometric features are derived from contents of zone, statistics, and font characteristics. Since text zones are characterized by the words they contain, word matching is an important operation in this module. For example, a zone has a higher probability of being labeled as “affiliation” when it contains words related to country, city, and school names. Also, a zone located between the words “abstract” and “keywords” has a higher probability of being labeled as “abstract” than other labels. For this purpose, fourteen word lists have been created, and the ternary search tree algorithm[56] is used as a search engine for the word matching.

The automated updating algorithm is based on a difference analysis between text zones generated by this subsystem and corresponding text zones corrected by operators during the reconcile stage. Statistical labeled text zone information obtained from this analysis can be used to improve the labeling accuracy.

Article zone contents modification server. This subsystem extracts, modifies, and reformats text in labeled zones according to MEDLINE database conventions. It consists of two modules: the *label cleanup* module and the *cleanup coach* module. The first prepares bibliographic record information for the reconcile operator, while the second operates on post-reconcile data to collect statistics in order to develop new rules to improve the performance of the first module.

The label cleanup module removes, replaces and/or processes HTML tags and special characters in labeled zones, and reformats the “title”, “author”, “affiliation” and “abstract” zones. Figure 12.2.3 shows examples of author and abstract before and after this process. The label cleanup module relies on rules derived by analyzing the page layout for each journal. Generic rules cover situations that occur in every journal, and for journals for which we have no a priori data. In addition, there will be rules specific for a particular journal. For unknown tags or confusing text, the results of any processing will be enclosed in special symbols such as curly brackets, { }, and the enclosed characters will be highlighted to alert reconcile operators.

Since rules implemented in the label cleanup module are used to prepare labeled zone contents for reconciliation by operators, the final corrected zone contents can be used to improve the module's performance. Given that a journal issue has been reconciled, the cleanup coach module automatically extracts differences between the final correct labels and the corresponding output from the label cleanup module. These differences may be analyzed manually or automatically to modify existing rules, or to infer additional rules for the label cleanup module.

MEDLINE article zones creation and Web-based reconciling workstation. This workstation selects and combines the appropriate portions of zone contents to create MEDLINE citation records for reconciling. The reconciling operator then checks, proofreads, searches, and corrects any remaining problems in the citation records including character and symbol errors, reformatting citations, capturing extra fields if necessary (pagination, data accession number, grant number, etc.), text zone labels selection and validation.

MEDLINE citation records uploading server. This subsystem creates an XML file based on NLMCommon DTD[46] for completed citation records of a journal issue and uploads the file to the NLM DCMS database for later indexing.

SGML-based MEDLINE citation records creation server. Some journal publishers also supply to NLM some portions of MEDLINE citation records such as "title", "author", "abstract", "pagination" in SGML format. However, the records are not always complete, so additional effort is often required to capture missing information. Since the SGML information even if incomplete is usually correct, this server subsystem exploits such information by generating text zones directly from it. These text zones will be used to further improve the labeling and reconciliation processes.

12.6 Summary

This section describes a proposed system to create biomedical bibliographic records for the MEDLINE database directly from online journals. The advantages of WebMARS over either manual keyboard entry or the scanning/OCR approach are saving labor costs and improving performance. The performance of our WebMARS prototype has been evaluated using a sample of 28 medical journal Web sites. Preliminary evaluation results show the feasibility of our design to create MEDLINE citations directly and effectively using Web document pages.

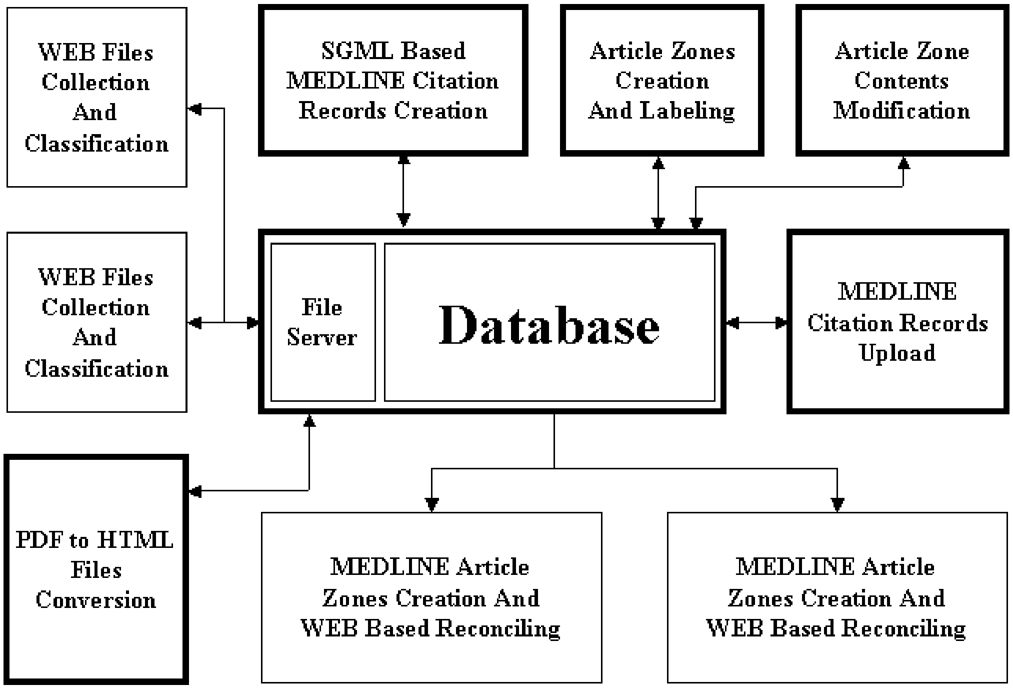

WEB Files Collection And Classification
SGML Based MEDLINE Citation Records Creation
Article Zones Creation And Labeling
Article Zone Contents Modification
WEB Files Collection And Classification
File Server
Database
MEDLINE Citation Records Upload
PDF to HTML Files Conversion
MEDLINE Article Zones Creation And WEB Based Reconciling
MEDLINE Article Zones Creation And WEB Based Reconciling

Figure 12.2.1: WebMARS system diagram (Bold outlines show servers)

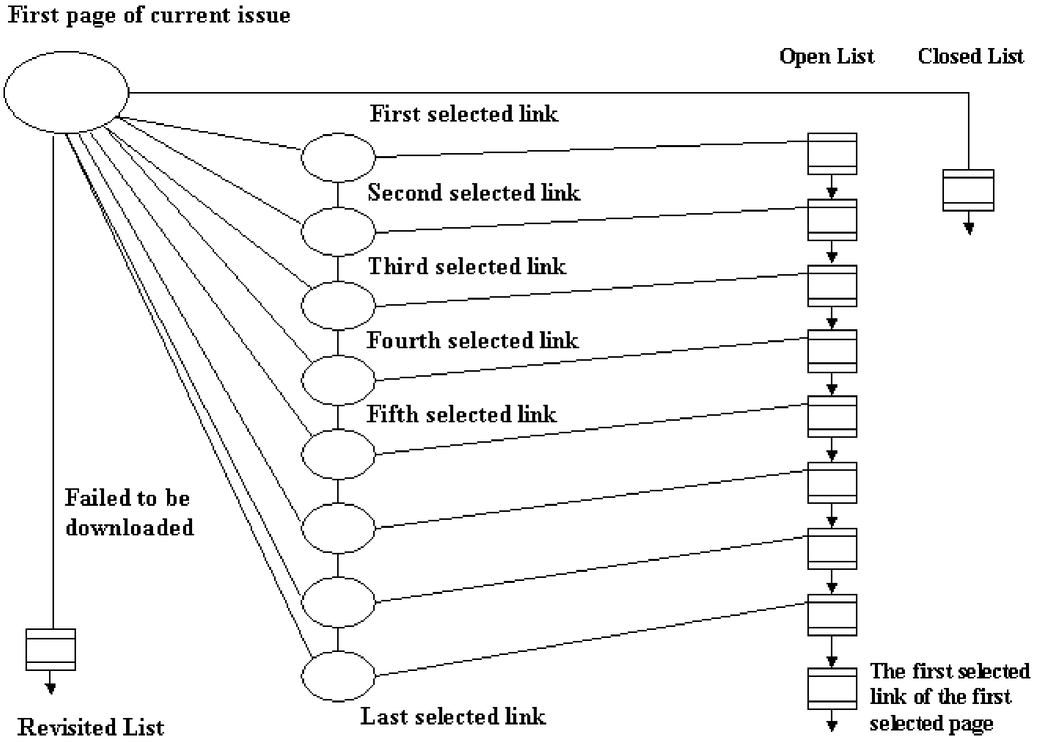


Figure 12.2.2 Dowloading process

Author label input:	**Author label output:**
<NOBR>Henry Abriel, MD, PhD</NOBR>	*Abriel H*
<NOBR>Michael V. Wehrens, MSc</NOBR>	*Wehrens MV*

Abstract label input	**Abstract label output**
<I>Background</I>&#151;Multiple mutations <I>SCN5A</I>, the gene that<SUP> </SUP> encodes the human Na<SUP>+</SUP> channel <IMG SRC="/math/agr.gif" ALT="{alpha}" BORDER="0">-subunit, are linked to 1 form…	*BACKGROUND: Multiple mutations of of SCN5A, the gene that encodes the human Na(+) channel alpha-subunit, are linked to 1 form*

Figure 12.2.3 Examples of author and abstract before and after the label cleanup process

12.3 Alternative method for text verification

The conventional approach to verifying the text output of any OCR system, or as in the case of MARS the output of a succession of automated processes, is to present the text in the same sequence as it appears on the printed page, and to highlight the low confidence characters (in color) in the text words. Then, as in our reconcile workstation, the operator can "tab" quickly from one suspected character to the next and make the necessary corrections. This conventional approach has some drawbacks. For example, the operator must detect the suspected character surrounded by a mass of correct text. Also, the text must be corrected as encountered, thereby breaking the rhythm of identifying incorrect characters.

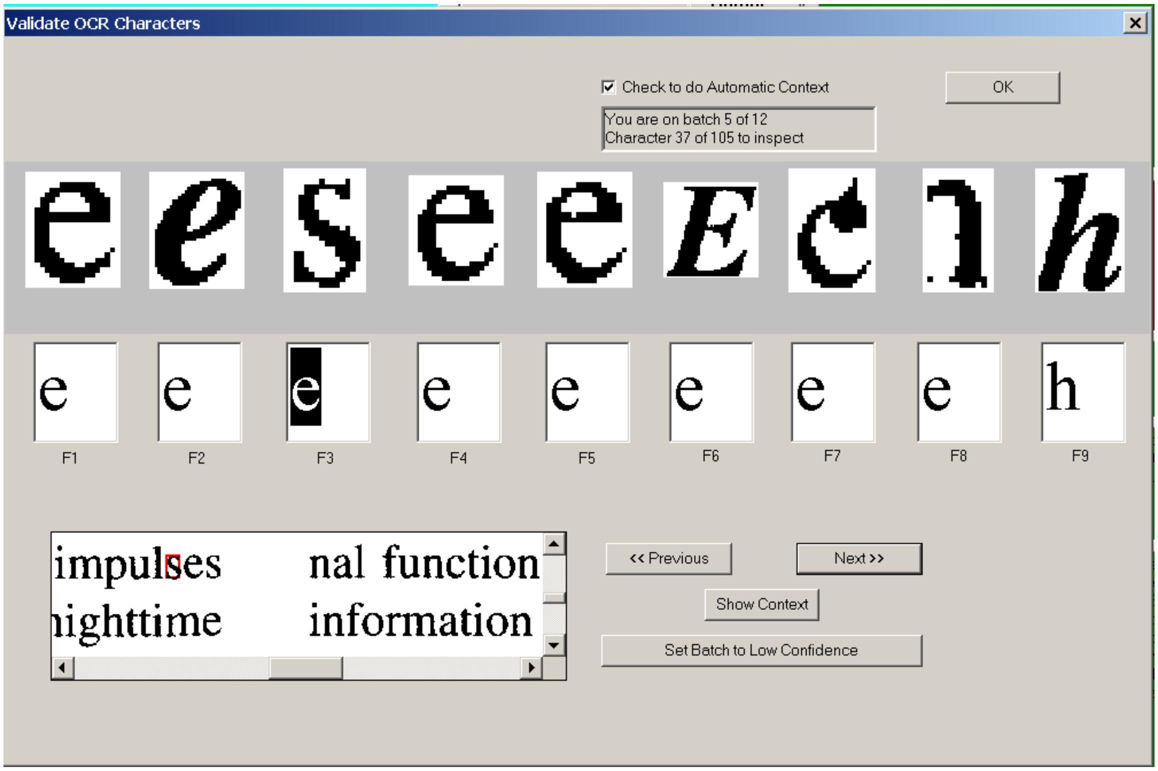
An alternative method is proposed that may prove to improve operator productivity. Called *Carpet Character Certification and Correction*, or the "Carpet" method, it involves grouping like characters (drawn from a number of pages or journal issues at the same time) and displaying them in groups in a single window, as shown in Figure 12.3.1. Each character appears in its "edit box." Only low confidence **A** - **Z** and **0 – 9** characters, of the same type, would be displayed in groups. The example shows a set of characters in the edit boxes, mostly **e'**s, some of them a misreading of an **s** or an **E** as shown in the corresponding bitmapped images right above the edit boxes. Since context is important to detect poorly captured character shapes, the system must provide the display of the image fragment (a word or phrase) that provides the context in which the (presumably) incorrect character appears. Such context will particularly help distinguish letters or numbers that appear similar, e.g., 1, I or 0,O.

The GUI for the Carpet system must possess the following functions:

1. An Automatic Context display of the TIFF image must be available when any of the edit boxes is focused on.
2. With a mouse click on an edit box, or using the F1 thru F9 keys, the operator should be able to correct the character. In Figure 12.3.1, the bitmap of each low confidence character appears above the edit text boxes. If the operator is unclear as to what the character is, he/she may type in a **'?'** that invokes the image fragment to provide the necessary context.
3. To continue, the operator should be able to select **Next>>** and load the next batch of 9 characters, or select **<<Previous** for a second look at the previous 9. By selecting Next>>, all characters are set to high confidence.
4. Reset all characters displayed to low confidence, in case of an accidental clicking of the Next>> button.
5. Click OK to stop processing.

The Carpet system is to be implemented using Visual C++ with the Kodak Image libraries. Following the software development, we intend to conduct a performance study using this system for reconciling, and measure the residual error rate and the time taken for correcting and verifying all the characters from a complete journal issue at a time. Should the accuracy and time saved prove to be an improvement over the current verification method, this module will be incorporated in the reconcile workstation software.

Figure 12.3.1 GUI for Carpet Character Certification and Correction



Validate OCR Characters

☑ Check to do Automatic Context

You are on batch 5 of 12
Character 37 of 105 to inspect

13 References

1. Thoma GR, Walker FL, Hauser SE. Biomedical document preservation by electronic imaging. In: Converting Information for WORM Optical Storage, Roth JP, ed., Meckler, 1990; 92-131.

2. Hauser SE, Thoma GR, Gass SI. Factors affecting the conversion rate of bound volumes to electronic form. Proc. Electronic Imaging '89, Boston MA, 1989; 1041-6.

3. Thoma GR, Hauser SE, Walker FL. Managing an archive of electronic document images. Proc. ASIS, vol. 26, 1989; 59-65.

4. Thoma GR, Cookson JP, Walker FL, Hauser SE. Interfacing optical disks to a document image storage and retrieval system. J. Imaging Technology, vol. 12, no. 5, October 1986; 288-92.

5. Thoma GR, Long LR, Berman LE. Design issues for a digital x-ray archive accessed over Internet. Proc. SPIE: Storage and Retrieval for Image and Video Databases II, vol. 2185; San Jose CA: 1994; 129-38.

6. Hauser SE, Hsu W, Thoma GR. Request routing with a back error propagation network. Proc. SPIE: Applications of Artificial Neural Networks IV, vol. 1965; Orlando FL: 1993; 689-95.

7. Thoma GR, Hauser SE, Le DX, Muller D, Walker FL. WILL: Advances in the management of interlibrary loan. Vol. 1. Internal technical report. Bethesda, MD: National Library of Medicine, Lister Hill National Center for Biomedical Communications, Communications Engineering Branch, September 1995; 26 pp.

8. Thoma GR, Hauser SE, Le DX, Muller D, Walker FL. WILL: Design of a standalone WILL unit. Vol. 2. Internal technical report. Bethesda, MD: National Library of Medicine, Lister Hill National Center for Biomedical Communications, Communications Engineering Branch, September 1995; 27 pp.

9. Walker F, Thoma GR. DocView: Providing access to printed literature through the Internet. Proc. 10th Integrated Online Library Systems Meeting, New York: Learned Information, Inc. May 1995; 165-73.

10. Thoma GR, Walker FL. Access to document images over the Internet. In: Rada R, Ghaoui C, eds. Medical Multimedia; Oxford UK: Intellect, 1995; 179-93.

11. Walker FL, Thoma GR. Multimode delivery of multimedia information. Proc. 14th National Conference on Integrated Online Library Systems, Medford NJ: Information Today Inc.; May 1999; 141-52.

12. Walker FL, Thoma GR. Access to document images over the Internet. Proc. 9th Integrated Online Library Systems Meeting, New York. May 1994; 185-197.

13. Le DX, Thoma GR, Wechsler H. Document image analysis using integrated image and neural processing. Proc. 3rd ICDAR'95, Montreal, Canada; Aug 1995; Vol. I, 327-30.

14. Le DX, Thoma GR, Wechsler H. Classification of binary document images into textual or non-textual data blocks using neural network models. Machine Vision and Applications, Issue 8, 1995, 289-304.

15. Le DX, Thoma GR, Wechsler H. Automated page orientation and skew angle detection for binary document images. Pattern Recognition, Vol. 27, No. 10, October 1994; 1325-44.

16. Le DX, Thoma GR. Automated portrait/landscape mode detection on a binary image. Proc. SPIE Symposium on Aerospace and Remote Sensing – Visual Information Processing II, Vol. 1961, Orlando FL, April 1993, 202-12.

17. Hauser SE, Cookson TJ, Thoma GR. Using back error propagation networks for automatic document image classification. Proc. SPIE Applications of Artificial Neural Networks IV, Vol. 1965, Orlando FL, April 1993, 142-50.

18. Le DX, Thoma GR. Document skew angle detection algorithm. Proc. SPIE Symposium on Aerospace and Remote Sensing – Visual Information Processing II, Vol. 1961, Orlando FL, April 1993, 251-62.

19. Le DX, Thoma GR, Wechsler H. Document classification using connectionist models. Proc. IEEE International Conference on Neural Networks, Orlando FL, June 1994, vol. 5; 3009-14.

20. Thoma GR, Le DX. Medical database input using integrated OCR and document analysis and labeling technology. Proc. 1997 Symposium on Document Image Understanding Technology, College Park, MD: University of Maryland Institute for Advances in Computer Studies; 1997; 180-81.

20a. Thoma GR. Automating data entry for an online biomedical database: a document image analysis application. Proc. 5th International Conference on Document Analysis and Recognition (ICDAR'99). Bangalore, India; Sept. 1999; 370-3.

21. Hauser SE, Browne AC, Thoma GR, McCray AT. Lexicon assistance reduces manual verification of OCR output. Proc. 11th IEEE Symposium on Computer-based Medical Systems. Los Alamitos, CA: IEEE Computer Society. June 1998; 90-5.

22. Le DX, Kim J, Pearson GF, Thoma GR. Automated labeling of zones from scanned documents. Proc. 1999 Symposium on Document Image Understanding Technology,

College Park, MD: University of Maryland Institute for Advances in Computer Studies; 219-26.

23. Ford GM, Hauser SE, Thoma GR. Automatic reformatting of OCR text from biomedical journal articles. Proc.1999 Symposium on Document Image Understanding Technology, College Park, MD: University of Maryland Institute for Advances in Computer Studies; 321-25.

24. Jain AK, Yu B. Document representation and its application to page decomposition. . IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, 1998, 294-308.

25. Nagy G, Seth S, Viswanathan M. A prototype document image-analysis system for technical journals. Computer, Vol. 25, 1992, 10-22.

26. O'Gorman L. The document spectrum for page layout analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, 1993, 1162-73.

27. Niyogi D, Srihari SN. Integrated approach to document decomposition and structual analysis. International Journal of Imaging Systems and Technology, Vol. 7, 1996, 330-42.

28. Farrow GSD, Xydeas CS, Oakley JP, Khorabi A, Prelcic NG. A comparison of system architecture for intelligent document understanding. Signal Processing-Image Communication, Vol. 9, 1996, 1-19.

29. Kanai J, Rice SV, Nartker TA, Nagy G. Automated evaluation of OCR zoning. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, 1995, 86-90.

30. Krishnamoorthy M, Nagy M, Seth S, Viswanathan M. Syntactic segmentation and labeling of digitized pages from technical journals. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, 1993, 737-47.

31. Baird HS. Model-directed document image analysis. Proc. 1999 Symposium on Document Image Understanding Technology, College Park, MD: University of Maryland Institute for Advances in Computer Studies, 42-9.

32. PrimeOCR Access Guide, Version 3.0. Prime Recognition, 1998.

33. Hones F, Lichter J. Layout extraction of mixed mode documents. Machine Vision and Applications 7, 1994, 237-46.

34. Taylor S, Fritzson R, Pastor J. Extraction of data from preprinted forms. Machine Vision and Applications 5, 1992, 211-22.

35. Tsujimoto S, Asada H. Major components of a complete text reading system. Proc. IEEE, Vol. 80, No. 7, 1992, 1133-49.

36. Tateisi Y, Itoh N. Using stochastic syntactic analysis for extracting a logical structure from a document image. Proc. IEEE Int. Conf. Neural Networks, Vol. 2, 1994, 391-94.

37. Niyogi D, Srihari SN. An integrated approach to document decomposition and structural analysis. Int. Journal of Imaging Systems and Technology, Vol. 7, 1996, 330-42.

38. Le DX, Thoma GR. Page layout classification technique for biomedical documents. Proc. World Multiconference on Systems, Cybernetics and Informatics (SCI 2000); Orlando, FL; Vol. 10, July 2000; 348-52.

39. Le DX, Thoma GR. Automated document labeling using integrated image and neural processing. Proc. World Multiconference on Systems, Cybernetics and Informatics, Orlando, FL; Vol. 6, 1999; 105-8.

40. Kim J, Le DX, Thoma GR. Automated Labeling in Document Images. Proc. SPIE: Document Recognition and Retrieval VIII, Vol. 4307, San Jose CA, January 2001, 111-22.

41. Hauser SE, Le DX, Thoma GR. Automated zone correction in bitmapped document images. Proc. SPIE: Document Recognition and Retrieval VII, Vol. 3967, San Jose CA, January 2000, 248-58.

42. Ford G, Hauser SE, Le DX, Thoma GR. Pattern matching techniques for correcting low confidence OCR words in a known context. Proceedings of SPIE, Vol. 4307, Document Recognition and Retrieval VIII, January 24-25, 2001, pp. 241-9.

43. Lasko TA, Hauser SE. Approximate string matching algorithms for limited-vocabulary OCR output correction. Proceedings of SPIE, Vol. 4307, Document Recognition and Retrieval VIII, January 24-25, 2001, pp. 241-9.

44. Pearson G, Moon CW. Bridging two biomedical journal databases with XML: A case study. Proc. 14th IEEE Symposium on Computer-Based Medical Systems. Los Alamitos, CA: IEEE Computer Society. July 2001, 309-14.

45. World Wide Web Consortium, XML Working Group, "XML 1.0 Specification 10-February-1998", ed. T. Bray, J. Paoli, C.M. Sperberg-McQueen. Available in various forms: www.w3.org/TR/1998/REC-xml-19980210.

46. "NLM to use XML and DTD for MEDLINE Data", 10/13/00, www.nlm.nih.gov/news/MEDLINEdata.html; the NLM Common DTD is given in www.nlm.hin.gov/bsd/licnesee.html.

47. Desai G and Fenner J. Unleash the power of XML. Imaging and Document Solutions, Vol. 9, No. 12, Dec 2000, 29-32.

48. Okun O, et al. An experimental tool for generating ground truths for skewed page images. Proc. SPIE Document Recognition and Retrieval VIII. Vol. 4307, San Jose CA, January 2001, 22-33.

49. Cha S-H, Srihari SN. Handwritten document image database construction and retrieval system. Proc. SPIE Document Recognition and Retrieval VIII. Vol. 4307, San Jose CA, January 2001, 13-21.

50. Doermann D and Mihalcik D. Tools and techniques for video performance evaluation. Proc. 15th International Conference on Pattern Recognition. Barcelona, Spain, September 2000, 167-70.

51. Swayne DF et al. Xgobi: interactive dynamic data visualization in the X window system. Journal of Computational and Graphical Statistics 7, 1998.

52. Barras C, et al. Transcriber: a free tool for segmenting, labeling and transcribing speech. Proc. 1st Int. Conf. Language Resources and Evaluation. Granada, Spain, May 1998; 1373-76.

53. Kanungo T, et al. TRUEVIZ: A groundtruth/metadata editing and visualizing toolkit for OCR. Proc. SPIE Document Recognition and Retrieval VIII. Vol. 4307, San Jose CA, January 2001, 1-12.

54. Le DX, Tran LQ, Chow J, Kim J, Hauser SE, Moon CW, Thoma GR. Automated medical records citation records creation for Web-based online journals. Proc. 14th IEEE Symposium on Computer-Based Medical Systems, Los Alamitos, CA: IEEE Computer Society. July 2001, 315-20.

55. Tran LQ, Moon CW, Le DX, Thoma GR. Web page downloading and classification. Proc. 14th IEEE Symposium on Computer-Based Medical Systems. Los Alamitos, CA: IEEE Computer Society. July 2001, 321-6.

56. Bentley J, Sedgewick B. Ternary search trees. Dr Dobb's Journal, April 1998, 20-25.

57. Hall PAV, Dowling GR. Approximate string matching. ACM Computing Surveys, 1980; 12:381-402.

58. Hauser SE, Sabir TF, Thoma GR. Speech recognition for program control and data entry in a production environment. Proc. SPIE: Intelligent Systems in Design and Manufacturing II, Vol 3833, 1999; 24-34.