



Politecnico di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

An automatic document processing system for medical data extraction

This is a pre-print of the following article

Original Citation:

An automatic document processing system for medical data extraction / Adamo, F.; Attivissimo, F; Di Nisio, A.; Spadavecchia, M.. - In: MEASUREMENT. - ISSN 0263-2241. - STAMPA. - 61:2(2015), pp. 88-99.
[10.1016/j.measurement.2014.10.032]

Availability:

This version is available at <http://hdl.handle.net/11589/6892> since: 2021-03-09

Published version

DOI:10.1016/j.measurement.2014.10.032

Terms of use:

(Article begins on next page)

An automatic document processing system for medical data extraction

Francesco Adamo, Filippo Attivissimo, Attilio Di Nisio, Maurizio Spadavecchia

Electrical and Electronic Measurements Laboratory

Department of Electrical and Information Engineering (DEI) – Politecnico di Bari

Via E. Orabona 4, 70125 Bari, Italy

[adamo, attivissimo, dinisio, spadavecchia]@misure.poliba.it

Abstract – *This paper illustrates an automatic document processing system for the extraction of data contained in medical laboratory results printed on paper. The final goal of the research is to automate the collection of medical data and to enable an efficient management and dissemination of the information. The following processing steps of the system are described in detail: image preprocessing; layout analysis for the identification of the tables contained in the document; extraction and classification of the laboratory results. Among the many features of the system there are the use of an open source OCR engine, as a basis of further processing, and the storage in XML format of the data retrieved, for ease of sharing. The knowledge base used to guide the data extraction is also explained. The proposed approach has been tested on several document formats and performance analyzed.*

Index terms – medical data, document image processing, medical services, performance evaluation

I. INTRODUCTION

Medical investigation is conducted to extend the life of people and to improve quality of life for patients with severe diseases; today it is common to run into doctors having different specializations which work together towards the common goal of curing a disease. This requires a multidisciplinary research organization utilizing advanced medical technologies and medical research institutes of different universities. Besides, in order to guarantee the statistical significance of the studies, a sufficient amount of data from clinical trials and medical examinations should be collected. Thus, the scientific communities of several medical fields are working on electronic databases containing clinical analyses and laboratory results useful for researches, medical investigations, epidemiological studies, quality control, and so on [1], [2], [3]. It should be stressed that an efficient and undistorted communication of medical research results and hospital data is one of the most important heritages of the medical scientific community.

As a matter of fact, however, a complete transition towards paperless practices has not been accomplished or, in some cases, is not possible at all, and paper continues to be used for diagnoses, laboratory results, and prescriptions. This constitutes an obstacle for the creation of electronic databases and electronic medical records [4]. Indeed, it has been noticed that the manual entry of data into medical records takes a long time and often produces errors [5], [6]. Moreover, the absence of common practices among various medical centers produces discrepancies and non-uniformity of data.

Therefore, the automatic conversion of paper documents into digital resources is an important and nontrivial task that greatly contributes to the preservation and dissemination of medical archives. In this paper an automatic system able to extract the data contained in tabular-like form

46 in printed medical laboratory results, converting them into an electronic form which can be stored
47 in databases and further processed is proposed.

48 A review of algorithms for the automatic analysis of printed documents is presented in Sec. II,
49 followed in Sec. III by a thorough description of the implemented system. Performance is
50 analyzed in Sec. IV and final remarks are given in Sec. V.

51

52

II. RELATED WORK

53 Extracting information from printed medical laboratory results in an automated way is not a
54 simple task, and requires several processing steps [3], [7], [8], [9]. Medical image archiving and
55 management allows a fast and objective diagnosis even from remote locations [10]. There are
56 many successful applications in this field [11], [12], [13], [14]. Document automation systems are
57 available for other purposes, such as generating special printed forms to be compiled by hand and
58 processed by OCR [15]. In [16] a method is proposed for the automatic text classification in
59 biomedical research documents, based on the use of a support vector machine. However, a
60 complete system tailored to laboratory results isn't available. In this work several components
61 have been integrated with the aim of building such a system. In order to better understand the
62 features of the proposed method, a short premise should be done about the components of the
63 system and the processing flow.

64 The main components required for processing the document are: digitization, pre-processing,
65 layout analysis, OCR, correction of the OCR results and document understanding. We consider
66 these ones as components rather than consecutive steps, because they can be used several times
67 with different working parameters in order to advance in the data extraction. For example, in our
68 proposed approach, layout analysis and OCR are iterated two times in order to find column

69 headers in the printed table of laboratory results. A third OCR run is performed with parameters
70 tailored to the contents expected in each table cell, which are predicted on the basis of the column
71 and the row where each cell is located.

72 Layout analysis allows to retrieve the structure of the document by using, essentially,
73 graphical features such as position, distance, orientation and size of the components being
74 analyzed, which can be connected components, characters, words, text lines, paragraphs, and so
75 on. Layout analysis has its roots in image segmentation algorithms, and is a fundamental step
76 towards document understanding, in which the logical relations between document components
77 are fully exploited. Image segmentation [17], [18] and text region extraction [19], [20], [21] are
78 one of the most debated issues in the document images analysis [16] and many problems are
79 currently unresolved. Over the last two decades, several techniques have been proposed, all
80 referable to three classes; bottom-up algorithms, top-down algorithms and hybrid algorithms [22].
81 In the bottom-up approaches text components are identified starting at the character level, then
82 characters are aggregated into words, and finally text lines, paragraphs and higher level
83 components are built to reassemble the whole pages; examples are the use of the Voronoi
84 diagram [23], the Docstrum algorithm [24], the Kruskal algorithm [25] and the probabilistic
85 approach [26]. Alternatively, in the top-down approaches, the pages are split into columns, then
86 into paragraphs and finally in the text lines and words. Examples are the XYcut [27] and
87 whitespace analysis [28]. Finally, hybrid approaches can be regarded as a mix of the above two
88 approaches in an attempt to overcome the limitations of these algorithms. Neural techniques have
89 been applied not only to OCR and word recognition, but also to layout analysis [29]. Due to the
90 importance of layout analysis in document image understanding, considerable effort has been
91 dedicated to the performance evaluation of these algorithms [16], [30]. No single algorithm can

92 be considered optimal and different approaches should be chosen depending on the specific
93 application.

94 In this work layout analysis is dedicated essentially to the analysis of tables, because this is the
95 form in which laboratory results are generally reported. Table detection can be performed by
96 analyzing gaps between words and rows, as described in [31]. However laboratory results often
97 are not tabulated in a strict manner, for example some cells can span more than one column, and
98 no cell box delimiters are printed. For this reason in our system a knowledge base is used, in
99 order to extract table data.

100 We have explored the possibility of integrating in our system an open source OCR software.
101 Really, any OCR could have been used because there aren't special requirements about advanced
102 functions and the layout analysis is performed mainly by our system. The only OCR features
103 used in this work, besides characters recognition, are the computation of the locations of the
104 characters and the restriction of the characters set (letters, digits, special symbols, and so on).
105 Matching scores for the recognized characters, which can be output by the OCR, are not used in
106 our system. Optical character recognition is prone to errors, and several techniques have been
107 proposed in literature to correct them [32]. This is useful in particular for the recognition of
108 handwritings. For example in [33] the use of specific language models is discussed, based on
109 statistical properties of words in text, morpho-syntactic characteristics of words and syntactic
110 structures of the language. In our application, which is aimed at the recognition of tables, we use
111 a different approach, based on a knowledge base of laboratory exams with their naming variants,
112 common measurement units, and approximate string matching algorithms. Our approach should
113 also be distinguished from other ones in which the knowledge-base contains only rules about the
114 geometric structure of the document [35]. We exploit the knowledge base and the logical
115 structure of the tables in order to correct OCR errors and recognize the laboratory exams, i.e. to

116 perform the document understanding. This method uses an explicit set of information to support
117 the data extraction so it requires some work in order to expand the range of recognized laboratory
118 exams but, for the same reason, it is relatively simple to implement and gives accurate results. In
119 particular, it has a low rate of matching errors as regards the recognition of the names of the
120 single exams contained in the document. Indeed, confusing one exam for another can be
121 dangerous in this kind of application.

122 As regards the general aspects of the processing flow, it should be noted that the extraction
123 and classification of data requires a lot of automatic decisions about where data are located and
124 what is their meaning, and each decision may have deep consequences on subsequent processing
125 [34]. This is in contrast with other industrial applications, where the authors have experienced
126 less correlation between subsequent processing steps [36], [36], [38], [39], [40]. Nonetheless in
127 the approach presented in this paper these decisions are taken sequentially and deterministically
128 evaluating best scores, and a careful choice of the algorithms employed avoids the need of going
129 back to reconsider past decisions. Even though walking backwards in the decision tree is a
130 possibility left open in our system, it has not been implemented because the achieved results,
131 reported in Sec. IV, are already satisfactory.

132

133

III. SYSTEM ARCHITECTURE

134 The conversion of paper and electronic documents into standard electronic forms is a key step
135 in medical research. There are many advantages in using standard electronic records: compact
136 and lossless storage, fast retrieval and transmission, easy data analysis and the possibility of
137 comprehensive statistical studies. Unfortunately, medical documents are very different in terms
138 of format, and medical lexicon is large; this leads to difficulties in the automatic creation or

139 conversion of records. In this paper we propose a method to automatically convert paper-based
140 medical reports into XML documents. Being easy to retrieve, display and index information
141 contained in XML documents, this will contribute to create a standard database useful to
142 researchers and clinicians.

143 The main stages that allow the processing of a printed page containing the laboratory results,
144 described in the following sub-sections, are: (Sec. A) image preprocessing, in which the
145 document readability is enhanced; (Sec. B) layout analysis, in which the document layout is
146 analyzed in order to identify columns and rows containing the information to be extracted, so
147 discarding extraneous graphical elements such as “margins” and graphs; (Secs. C and D) data
148 extraction and classification, in which text returned by the OCR is analyzed syntactically and
149 semantically; (Sec. E) exportation in XML format of the extracted data. A knowledge base (KB),
150 which assists layout analysis, data extraction and classification, is described in Sec. F.

151 *A. Preprocessing*

152 In this phase the image is prepared for subsequent processing steps. In particular, equalization,
153 binarization and suppression of long lines are required to ease layout analysis and OCR, and to
154 remove various artifacts which could give rise to misinterpretations. Let $\mathbf{A}_0 = [A_0(i, j)]$ be the
155 gray-scale input image. An example image, obtained with a common scanner with resolution 300
156 dpi, is shown in Fig. 1. The procedure is as follows.

- 157 a) In a first step, if the skew angle is greater than 1.0° , \mathbf{A}_0 is deskewed, obtaining the image \mathbf{A}_1 .
- 158 b) \mathbf{A}_1 is equalized in order to cover the full 256 gray-levels range, so obtaining the image \mathbf{A}_2 .
- 159 c) \mathbf{I}_3 is then calculated by binarizing \mathbf{A}_2 with a combination of thresholding techniques. In
160 particular, a first image \mathbf{I}_1 is obtained with a threshold t_1 fixed at the 90% of the intensity
161 range and inversion:

$$I_1(i, j) = \begin{cases} 0, & \text{if } A_2(i, j) > t_1 \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

with $t_1 = 0.9 \times 255$.

A second image \mathbf{I}_2 comes from adaptive thresholding on \mathbf{A}_2 and inversion:

$$I_2(i, j) = \begin{cases} 0, & \text{if } A_2(i, j) > T_2(i, j) - c_2 \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

where $T_2(i, j)$ is the mean of the $N_2 \times N_2$ neighbourhood of $A_2(i, j)$, with $N_2 = 7$ and $c_2 = -35$.

Finally, \mathbf{I}_3 is given by

$$\mathbf{I}_3 = \mathbf{I}_1 \cap \mathbf{I}_2 \quad (3)$$

d) Image \mathbf{I}_5 is obtained by deleting long horizontal and vertical lines, in order to avoid interference with OCR. Firstly, the thresholded image \mathbf{I}_4 is prepared,

$$I_4(i, j) = \begin{cases} 0, & \text{if } A_2(i, j) > T_4(i, j) - c_4 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where $T_4(i, j)$ is the mean of the $N_4 \times N_4$ neighbourhood of $A_2(i, j)$, with $N_4 = 17$ and $c_4 = 6$. The lines are detected by applying to \mathbf{I}_4 the algorithm discussed next, so obtaining the image \mathbf{I}_L which contains only horizontal and vertical lines. These lines are removed by performing the set difference

$$\mathbf{I}_5 = \mathbf{I}_3 - \mathbf{I}_L \quad (5)$$

When horizontal lines are being detected on the binary image \mathbf{I}_4 , the following operations are accomplished:

- Connected components whose area is less than 2 pixels are filtered out.

- 180 - Morphological closure is performed with a horizontal structuring element of length 15 pixels,
 181 followed by morphological opening with a horizontal structuring element of length 20 pixels.
- 182 - The following characteristics are calculated for each connected component of the resulting
 183 image: minimum width wm in the vertical direction, maximum width wM in the vertical
 184 direction, length len in the horizontal direction. All characteristics are measured in pixels.
- 185 - A component is recognized as a line iff

$$186 \quad len \geq 50 \text{ and } \frac{len}{wM} > 30 \text{ and } \frac{wM - wm}{len} < 0.01 \quad (6)$$

187 The detection of vertical lines proceeds analogously.

188 e) Finally small size components are filtered out,

$$189 \quad \mathbf{I}_6 = F[\mathbf{I}_5] \quad (7)$$

190 where the operator F filters connected components whose area is less than 3 pixels. The area is
 191 calculated by using the Green's formula applied to the component contours[41], as implemented
 192 in OpenCV (Open Source Computer Vision Library) [42].

193 ***B. Layout Analysis***

194 In the layout analysis phase, the image is subdivided into blocks; text rows are processed by the
 195 OCR, and the text is compared with a list of column headers. The column headers identify those
 196 table columns which contain pertinent medical data. Those found by the algorithm in the example
 197 image of Fig. 1 are enclosed in red boxes in Fig. 2 and Fig. 3.

198 Successively, the rows are analyzed in order to identify the table cells and the semantic of words
 199 and group of characters. A distinction is performed between test names (enclosed in blue boxes in
 200 Fig. 2 and Fig. 3, numerical and alphanumeric results (green boxes), measurement units

201 (magenta boxes) and reference values (yellow boxes). Tests that are preceded by a header rows
202 and multi-row tests, which extend over multiple rows, are also identified. Header rows are
203 enclosed in blue boxes, as test names, in Fig. 2 and Fig. 3.

204 The algorithm is detailed in the remainder of this paragraph. The input image \mathbf{I}_6 is the one
205 obtained at the end of the pre-processing.

206

207 *Segmentation of table headers*

208 a) A first segmentation of document rows is performed with the aim of identifying which one
209 contains table headers. The segmentation is performed with the following steps: horizontal
210 projection, so obtaining vertical regions; zero-padding of size 3 at both ends of the
211 projection; morphological opening with a structuring element of size 6; deletion of the zero-
212 padding; filling of empty regions of size 1 pixel; the resulting vertical regions, expanded
213 horizontally up to the size of \mathbf{I}_6 , constitute the binary mask which denotes document rows.

214 b) The rows are processed individually by the Tesseract OCR[43]. Let r_k be the text contained
215 in the k -th row.

216 c) The previously processed rows are searched for columns headers by means of approximated
217 string matching. Four column types are considered: test name, test result, test unit, test
218 reference range. For each column type, a list of headers text variants is stored in the KB. Let
219 h_{ij} be the j -th text variant associated with the i -th column type, where $i=1, \dots, N_C$ and $N_C = 4$.
220 To find the row containing the column headers, firstly the Levenshtein distance (with
221 deletion, insertion and substitution costs all equal to 1) between string h_{ij} and its best
222 matching substring in r_k is computed and denoted as l_{ijk} . Then a cost function $c_{ijk} =$
223 $l_{ijk}/\text{len}(h_{ij})$ is calculated, where the function len returns the number of characters of its

224 argument. The lower cost text variant for each column class is calculated as $c'_{ik} =$
225 $\operatorname{argmin}_j c_{ijk}$. The value c'_{ik} is calculated for each row k . A header row K is found if $\sum_i c'_{iK}/$
226 $N_C \leq 0.34$ and $c'_{iK} \leq 0.7$ for each i . If there isn't one and only one such row, then an error is
227 returned.

228 d) The coordinates of the boxes enclosing the column headers are calculated by examining the
229 headers row K . Hence, for each column type i , the selection of the more appropriate text
230 variant j is refined by using a new cost function $c''_{ij} = l_{ijK}/\operatorname{len}(h_{ij}) - 1.5 \cdot \operatorname{len}(s_{ij})/$
231 $\operatorname{len}(r'_K)$, where r'_K is obtained from r_K by removing spaces, and s_{ij} is the substring of r'_K that
232 best matches h_{ij} . After the text of each column headers is identified, the enclosing boxes are
233 finally individuated.

234

235 *Segmentation of table cells*

236 To segment table cells in a robust manner, at the beginning only the text contained in the vertical
237 projection of the column headers is taken into account (step e), then the cells are extended
238 horizontally and vertically to include nearest text (step f). The fact that table cells can have
239 variable sizes help in filtering out irrelevant graphical elements.

240 e) The table rows under the column headers are segmented as follows. Firstly, the pixels in the
241 image \mathbf{I}_6 that are not under the boxes enclosing the column headers are zeroed, so obtaining
242 the image \mathbf{I}_7 . A binary mask \mathbf{I}_8 is then calculated by applying to \mathbf{I}_7 the same procedure
243 described previously in step a). Finally, the table rows are obtained by masking \mathbf{I}_6 with \mathbf{I}_8 .

244 f) Table cells in the previously found table rows are segmented by analyzing blanks with the
245 following procedure. Each row separately is subjected to vertical projection, zero-padding of
246 size 1 at both ends of the projection, morphological opening with a structuring element of

size 2, deletion of the zero-padding, and filling of empty regions of size less than 20 pixel. This gives, for each row, the horizontal extension of each table cell. The vertical extensions of table cells are expanded by considering the connected components of the horizontal projection of I_6 . In particular, each cell is vertically extended up to the size of the connected components that intersects it when the block is horizontally projected onto the y -axis.

Classification of cells

g) Table cells are tagged with the column types they belong to. For any given row, this classification procedure considers in turn each column type and tries to tag the cell that better corresponds to that column. Column types are examined with the following priority order: test name, test result, test unit, test reference range. The condition for tagging a table cell is that it should overlap with the column header box after projection on the x -axis and should not have been previously associated with another column type. If more cells overlap with the same header, then the wider cell is selected.

The rows are classified according to the cell tags they contain, following the rules described in *Table I*. Four classes are defined: test row, name row, result row, invalid row. A row is invalid if none of the rules specified in *Table I*

h) applies. This classification is based only on the layout analysis of rows one at time, while the logical relations among rows are detected in the data extraction phase, where KB is also used.

C. Data extraction

In this phase data are extracted from table cells.

269 a) The cells are processed by the OCR, which is configured to recognize a different set of
270 characters according to the cell type. If a test result cell and a test unit cell are in the same
271 row, then the set of characters recognized in the test result cell are those used to represent
272 numbers.

273 b) The text retrieved by the OCR is then analyzed in order to recognize the laboratory tests it
274 contains. In particular the data classification routine (DCR), which is described in the
275 following section, is applied to *name rows* and *test rows*. The DCR is aimed at determining
276 the test definition in the KB that matches the table row, that this to classify that row, and
277 extract the relevant information. The KB contains the list of all medical tests (LMT) that the
278 system is able to recognize, with details such as test names (including variants) and
279 measurement units. It should be also taken into account that header rows can occur in a
280 document to put in evidence the logical structure of the document, or to better specify the test
281 name cell contained in another row when ambiguities can result. Hence the KB contains also
282 a list of headers (LH), where each header can be of one of two types denoted as *H2* and *H1*.
283 When *test rows* are analyzed, the DCR uses the LMT and the LH, while *name rows* are
284 analyzed by using only the LH. Each test name variant specified in the LMT can be
285 optionally associated (by the KB) with a header belonging to the LH.

286 When a table row contains a test name variant associated with an H2 header, this means that
287 the table row should appear in a row after that header (separated by zero or more other rows),
288 and otherwise it is discarded. Instead, if a test name variant is associated with an H1 header,
289 then this header should appear immediately before that table row (separated by zero other
290 rows), otherwise it is discarded. To allow more flexibility in data extraction, *test rows* which
291 are recognized as H1 headers are considered valid even if they are not immediately followed
292 by an associated row. In other words, H1 header rows can also carry complete test

293 information, while H2 headers should be necessarily associated with other rows in order to
294 be interpreted as meaningful information. Moreover, if the table row contains a *test name* but
295 not a *result cell*, then the result and/or measurement unit are eventually taken from the
296 corresponding cells in the associated H1 header row.

297 Finally, the case is also considered in which the result is multi-row. This happens, for
298 example, when the result is in the form of descriptive text spanning more rows, which are
299 expected to be classified as *result rows*. These rows are processed by the DCR only if the
300 preceding row is recognized in the KB as being part of a multi-row result, and in that case
301 the test results from different rows are aggregated.

302 c) Special rules are applied selectively to some recognized tests, according to the KB. For
303 example, categorical results are matched against a list of known words.

304 d) For each recognized test, a record is created which contains the raw OCR data, and the data
305 normalized with the help of the KB. The normalized data include a unique test identifier, a
306 standardized name (chosen among different variants that can occur in medical laboratory
307 results), the result and, eventually, the measurement unit.

308 ***D. Data classification routine (DCR)***

309 This section describes the data classification routine, which is called during the data extraction
310 phase reported in the previous section.

311 In the following discussion, the test definitions are individuated by index i , with $i = 0, \dots, I$.
312 For simplicity, we assume here that naming variants of the same medical test count as separate
313 entries in the KB.

314 Let $[k_i]$ be the list of test names and/or headers retrieved from the KB, and s be the string
315 contained in the test name cell of the table row under analysis. For each string k_i in the list, the

316 substring s'_i of s best-matching k_i is found by considering the Levenshtein distance, with
 317 deletion, insertion and substitution costs all equal to 1. Let l_i be the Levenshtein distance between
 318 s'_i and k_i . For each string k_i in the list, a cost is then calculated as

$$319 \quad c_i = \left[l_i + \frac{\text{len}(k_i) - \text{len}(s'_i)}{\text{len}(s'_i)} + \frac{\text{len}(s) - \text{len}(s'_i)}{\text{len}(s'_i)} \right] / \text{len}(k_i) \quad (8)$$

320 If the row being analyzed isn't a test row, then the classification proceeds by calculating $i' =$
 321 $\text{argmin}_i c_i$.

322 Otherwise the three KB definitions that correspond to the least costs, denoted as $[c_j]$ with $j \in$
 323 J , are further processed by taking into account measurement units. The string contained in the
 324 measurement unit cell, which can eventually be empty, is parsed by eliminating spaces and the
 325 'x' multiplier symbol, if they are present. At this point the measurement unit string is compared
 326 with a list of possible measurement units (including variants) described in the test definition j .
 327 Let l'_j be Levenshtein distance of the best matching measurement unit. If no measurement unit is
 328 specified in the KB, then l'_j is zero. A cost l''_j is then calculated as $l''_j = c_j + l'_j / 3$, and finally
 329 the row is classified with the KB definition $i' = \text{argmin}_{j \in J} l''_j$.

330 If the classification has cost $c_{i'}$ greater than 0.51, then the procedure fails and the row is
 331 discarded.

332 The content of the *result cell* is stored internally as a floating point number, if possible.
 333 Otherwise it is interpreted as a generic alphanumeric string.

334 ***E. Exportation in XML format***

335 In this phase, the data extracted from the document are saved in an XML output file. An
 336 excerpt of a typical output file is shown in Fig. 4.

337 The most relevant elements in the XML output have tag <test>. A <test> element contains the
338 following elements:

- 339 - <testId> and <name> are, respectively, the unique test identifier and the test name. The
340 available identifiers and test names are listed in the KB.
- 341 - <result> and <unit> are the result and the measurement unit of the test.
- 342 - <nameRaw>, <resultRaw>, <unitRaw> are the raw results of the OCR (without KB-
343 driven post-processing), indicating respectively the test name, the test result, and the
344 measurement unit.
- 345 - <nameIm>, <resultIm>, <unitIm>, are the coordinates of the boxes which bound the
346 corresponding elements in the image. The attribute *array* indicates the size, 1 x 4, of the
347 array containing the coordinates. This array is serialized to a string that can be easily
348 parsed.

349 The previously mentioned elements have an attribute, *type*, with values “numeric” or “string”,
350 that indicate the class of the data contained inside the element.

351 This XML output has been conceived in order to share medical laboratory results in a simple
352 manner.

353 ***F. Knowledge base (KB)***

354 The KB contains the definitions of the medical tests and the test headers that the system is able
355 to recognize. For simplicity, each definition corresponds to a table row in a spreadsheet file, an
356 excerpt of which is given in Table II.

357 Each definition consists of a set of attributes:

- 358 - *testId*: unique test or header identifier;
- 359 - *name*: normalized test name;

- 360 - *unit*: normalized test unit;
- 361 - *nameAliases*: list of test name variants;
- 362 - *unitAliases*: list of measurement unit variants;
- 363 - *testHeaderId*: list of *id* headers associated to test name variants;
- 364 - *headerType*: specifies if it is an header, and which type of header (H1 or H2);
- 365 - *multiRow*: specifies if the result is multi-row;
- 366 - *specialParsing*: specifies post-processing functions for handling particular tests.

367

368 Since medical tests may be given different names in different laboratories, it is necessary to
369 relate each variant to a unique test identifier (*testId* attribute) and descriptive name (*name*
370 attribute). The possible textual variants of the test name, which comprise also abbreviations and
371 different spellings, are collected in a list (*nameAliases* attribute). The same reasoning applies to
372 normalized units (*unit* attribute) and their variants (*unitAliases* attribute). The use of the KB is
373 particularly useful for the correct recognition of the measurement units, because OCRs, if not
374 properly trained or configured, can have high error rates due to the presence of non-Latin
375 characters (e.g. μ) that are wrongly recognized.

376 Three other attributes, *testHeaderid*, *headerType*, *multiRow*, allow the system to handle cases
377 in which the data relevant to a given test should be extracted by examining multiple lines. For
378 any given name variant of a medical test, an associated header can eventually be specified by
379 using the *testHeaderId* attribute. For header definitions, the *headerType* attribute differentiates
380 between H1 and H2 headers, whose meanings have been explained in Sec. III.C. The *multiRow*
381 attribute is used to specify that the result can occupy more than one row as happens, for example,
382 for the microscopic analysis of the urine sediment, which often occurs in form of some
383 descriptive text.

384 The *specialParsing* attribute is aimed at handling tests, such as *Urine specific gravity*, in
385 which the measurement unit is often not given and should be deduced by the numerical result
386 choosing among ‘g/mL’ and ‘g/L’.

387

388

IV. SYSTEM PERFORMANCE

389

390 In this Section experimental results are illustrated. The system performance has been analyzed
391 in detail by applying the proposed method to the recognition of printed laboratory tests, and
392 quantitative results are reported.

393 The KB used in this experimentation included about 120 definitions, tailored for recognizing
394 four different kinds of documents coming from different laboratories. These documents contained
395 laboratory test prescribed regularly to patients by nephrologists. The algorithm parameters were
396 chosen by means of experimentation on a few pages.

397 The final evaluation procedure consisted in the following steps:

- 398 - Selection of 20 pages of laboratory tests, in Italian language. Each page, whose size is A4,
399 has been digitalized at 300 dpi by an off the shelf scanner, giving grayscale image files in
400 PNG format.
- 401 - Creation of a ground-truth database containing the medical tests present on each page.
402 Each record, which will be referred later as *true test*, contains the following fields: *testId*,
403 normalized test *name*, test *result*, normalized measurement *unit*, page identifier *pageId*.
404 The ground-truth database contained 480 *true tests*.
- 405 - Processing of the images with the proposed system, and creation of output XML files
406 containing the extracted data. Creation of a database of extracted data, with the same

407 structure of the ground-truth database, containing what will be referred as *estimated tests*
408 in the following discussion. Image skew was not corrected.

- 409 - Comparison between *true tests* and *estimated tests*, and automatic creation of a report.

410
411 Total processing time was about 18 minutes using a notebook with an Intel Core 2 X9100 3
412 GHz processor and 4 GB of RAM.

413 Global statistics relevant to the entire set of documents are illustrated in Table III.

414 The definition of each event counted by the statistics and their relative frequency of
415 occurrence (obtained dividing by the number of *true tests*) are reported.

416 The *test mismatch* error rate, which counts *true tests* that doesn't match any *estimated test* on
417 the same page, is 4.8%.

418 *Test name mismatches* were almost due to excessively bad character recognition, with only
419 one case of wrong segmentation in which the test *name* and the test *result* were merged in a
420 single cell. Hence, the algorithm was selective in discarding test *names* recognized with too many
421 character errors, according to the threshold illustrated in Sec. III.D .

422 Two *numeric mismatches* were due to the decimal separator (a comma in Italian) interpreted
423 as a digit.

424 Two *string mismatches* occurred. In the first case, an anomalous result, ">400", has been
425 extracted as "5400". In the second case, a long alphanumeric result has been extracted with some
426 OCR errors, but it is still readable (in Italian): "ESAME MICROSCOPICO DEL SEDIMENTO
427 Alcune cellule basse vie;0-5 Leucciti per campo;0-5 Emazie per campo;Diversi cristalli ossalate
428 di calcio".

429 It should be noted that the *spurious test names* statistic is zero, meaning that the system has
430 never interpreted some characters as tests when they are not.

431 This is true also for a page, included in the experiment, which contained only the laboratory
432 template and no tests to extract.

433

434

V. CONCLUSION

435 The manual insertion of laboratory results by the medical or nursing staff is time consuming
436 and produces errors. However the use of electronic medical records and databases has many
437 benefits, among which the possibility of improving treatments of diseases more readily and
438 accurately, and the availability of clinical data for research purposes.

439 The algorithm presented here, which has been experimented on laboratory results of patients
440 with renal problems, overcomes the limitations of manual entering and is able to extract and
441 interpret data originated from different laboratories.

442 This kind of automation is not a simple task, as it requires many processing steps and several
443 parameters to be tuned. However the fact that the problem has been subdivided in to steps with
444 fine granularity simplifies the tuning of the system and make the algorithm applicable, in
445 prospective, to a large set of typologies of laboratory results.

446

447

ACKNOWLEDGEMENT

448 The Authors would like to thank the technical staff of ApuliaBiotech for providing the printed
449 laboratory results.

450

451

REFERENCES

452 [1] N. Black, S. Tan, Use of National Clinical Databases for Informing and for Evaluating
453 Health Care Policies, Health Policy, vol. 109, pp. 131-136, February 2013.

- 454 [2] C. Zoccali, Clinical Databases in Nephrology: Research and Clinical Practice Goals
455 and Challenges, *Journal of Nephrology*, vol. 19, pp. 551-555, October 2006.
- 456 [3] M. R. Ogiela, R. Tadeusiewicz, Nonlinear Processing and Semantic Content Analysis
457 in Medical Imaging - A Cognitive Approach, *IEEE on Instrumentation and*
458 *Measurement*, vol. 54, no. 6 , pp. 2149-2155, December 2005.
- 459 [4] F. Adamo, F. Attivissimo, A. Di Nisio, An Integrated System for the Management of
460 Medical Data, *Proc. of MeMeA IEEE Symposium*, May 30-1, 2011, Bari, Italy, pp.
461 241-243.
- 462 [5] S. I. Goldberg, A. Niemierko, A. Turchin, Analysis of Data Errors in Clinical Research
463 Databases, *Proc. of Annual AMIA Symposium*, October 22-23, 2008, pp. 242–246.
- 464 [6] J. T. Scott, T. G. Rundall, T. M. Vogt, J. Hsu, Kaiser Permanente's Experience of
465 Implementing an Electronic Medical Record: a Qualitative Study, *British Medical*
466 *Journal*, vol. 331, pp. 1313–1316, December 2005.
- 467 [7] F. Russo, A Method based on Piecewise Linear Models for Accurate Restoration of
468 Images Corrupted by Gaussian Noise, *IEEE on Instrumentation and Measurement*, vol.
469 55, no. 6 , pp. 1935-1943, December 2006.
- 470 [8] L. Cinque, S. Levialedi, L. Lombardi, S. Tanimoto, Segmentation of page images
471 having artifacts of photocopying and scanning, *Pattern Recognition*, vol. 35, pp. 1167-
472 1177, 2002.
- 473 [9] R. Gupta, J. N. Bera, M. Mitra, Development of an embedde system and Matlab-based
474 GUI for online acquisition and analysis of ECG signal, *Measurement*, vol. 43, pp.
475 1119-1126, 2010.
- 476 [10] M. Engin, E.Caglav, E. Z. Engin, Real-time ECG signal transmission via telephone
477 network, *Measurement*, vol. 37, pp. 167-171, 2005.
- 478 [11] S. G. Mougiakakou, I. k. Valavanis, N. A. Mouravliansky, A. Nikita, K. S. Nikita,
479 Diagnosis: A Telematics-Enabled System for Medical Image Archiving, Management,
480 and Diagnosis Assistance, *IEEE on Instrumentation and Measurement*, vol. 55, no. 6,
481 pp. 2113-2120, July 2009.

- 482 [12] C. De Capua, A. Menduri, R. Morello, A Smart ECG Measurement System based on
483 Web-Service-Oriented Architecture for Telemedicine Applications, IEEE on
484 Instrumentation and Measurement, vol. 59, no. 10, pp. 2530-2538, July 2010.
- 485 [13] J. Gilchrist, M. Frize, C. M. Ennett, E. Bariciak, Performance Evaluation of Various
486 Storage Formats for Clinical Data Repository, IEEE on Instrumentation and
487 Measurement, vol. 60, no. 10, pp. 3244-3252, October 2011.
- 488 [14] M. Ceccarelli, A. Speranza, D. Grimaldi, F. Lamonaca, Automatic Detection and
489 Surface Measurements of Micronucleus by a Computer Vision Approach, IEEE on
490 Instrumentation and Measurement, vol. 59, no. 9, pp. 2383-2390, September 2010.
- 491 [15] H. Fujisawa, Forty years of research in character and document recognition-an
492 industrial perspective, Pattern Recognition, vol. 41, pp. 2435-2446, 2008.
- 493 [16] M. Berardi, M. Ceci, D. Malerba, A Hybrid Strategy for Knowledge Extraction from
494 Biomedical Documents, Proc. of NNLDAR Symposium, October 22-23, 2005, Seoul,
495 Korea, pp. 18-22.
- 496 [17] Y. Wang, I. T. Phillips, R. M. Haralick, Document zone content and its performance
497 evaluation, Pattern Recognition, vol. 39, pp. 57-73, 2006.
- 498 [18] P. D. Sathya, R. Kayalvizhi, Amended bacterial foraging algorithm for multilevel
499 thresholding of magnetic resonance brain images, Measurement, vol. 44, pp. 1828-
500 1848, 2011.
- 501 [19] Y. Xiao, H. Yan, Text region extraction in a document image based on the Delaunay
502 tessellation, Pattern Recognition, vol. 36, pp. 799-809, 2003.
- 503 [20] A. Antonacopoulos, S. Pletschacher, D. Bridson, C. Papadopoulos, ICDAR 2009 Page
504 Segmentation Competition, Proc. of ICDAR 2009, July 26-29, 2009, Barcelona,
505 Spain.
- 506 [21] P. A. Belan, S. A. Araujo, A. F. H. Librantz, Segmentation-free approaches of
507 computer vision for automatic calibration of digital and analog instruments,
508 Measurement, vol. 46, pp. 177-184, 2013.
- 509 [22] R. Cattoni, T. Coianiz, S. Messelodi, CM Modena, Geometric Layout Analysis
510 Techniques for Document Image Understanding: a Review, ITC-IRST Technical
511 Report, TR9703-09, pp. 1-68, 1998.

- 512 [23] K. Kise, A. Sato, M. Iwata, Segmentation of Page Images Using the Area Voronoi
513 Diagram, *Computer Vision and Image Understanding*, vol. 70, pp. 370-382, September
514 1998.
- 515 [24] L. O’Gorman, The Document Spectrum for Page Layout Analysis, *IEEE on Pattern
516 Analysis and Machine Intelligence*, vol. 15, pp. 1162-1173, September 1993.
- 517 [25] A. Simon, J. C. Pret, A. P. Johnson, A fast algorithm for bottom-up document layout
518 analysis, *IEEE on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 273-
519 277, Mar 1997.
- 520 [26] J. Liang, I. T. Phillips, R. M. Haralick, An optimization methodology for document
521 structure extraction on Latin character documents, *IEEE on Pattern Analysis and
522 Machine Intelligence*, vol. 23, no. 7, pp. 719,734, Jul 2001.
- 523 [27] G. Nagy, S. Seth, M. Viswanatham, A Prototype Document Image Analysis System for
524 Technical Journal, *Computer*, vol. 25, pp. 10-22, July 1992.
- 525 [28] H. S. Baird, Background structure in Document Images, *Document Image Analysis*,
526 vol. 15, pp. 17-34, July 1994.
- 527 [29] S. Marinai, M. Gori, G. Soda, Artificial neural networks for document analysis and
528 recognition, *IEEE on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 23-
529 35, Jan. 2005.
- 530 [30] F. Shafait, D. Keysers, T. M. Bruel, Performance Evaluating and Benchmarking of
531 Six-Page Segmentation Algorithms, *IEEE on Pattern Analysis and Machine
532 Intelligence*, vol. 30, pp. 941-954, June 2008.
- 533 [31] S. Mandal, S. P. Chowdhury A. K. Das, B. Chanda, A Simple and Effective Table
534 Detection System from Document Images, *Journal of Document Analysis*, vol. 8, pp.
535 172-182, September 2006.
- 536 [32] M. Malburg, Comparative Evaluation of Techniques for Word Recognition
537 Improvement by Incorporation of Syntactic Information, *Proc. of ICDAR 1997*,
538 August 18-20, 1997, Ulm, Germany.
- 539 [33] M. Piasecki, Multilevel correction of OCR of medical texts, *Journal of Medical
540 Informatics and Technologies*, vol. 11, pp. 263-273, November 2007.

- 541 [34] K. C. Fan, L. S. Wang, Classification of document blocks using density features and
542 connectivity histogram, Pattern Recognition letter, vol. 16, pp. 952-962, 1995.
- 543 [35] K. H. Lee, Y. C. Choy. S. B. Cho, Geometric structure analysis of document images: a
544 knowledge-based approach, IEEE Pattern Analysis and Machine Intelligence, vol. 22,
545 no. 11, pp.1224-1240, Nov 2010.
- 546 [36] F. Adamo, F. Attivissimo and A. Di Nisio, Calibration of an inspection system for
547 online quality control of satin glass, IEEE on Instrumentation and Measurement, vol.
548 59, no. 5, pp. 1035-1046, May 2010.
- 549 [37] F. Adamo, F. Attivissimo, A. Di Nisio and M. Savino, A low-cost inspection system
550 for online defects assessment in satin glass, Measurement, vol. 42, no. 9, pp. 1304-
551 1311, November 2009.
- 552 [38] F. Adamo, F. Attivissimo, A. Di Nisio and M. Savino, An online defects inspection
553 system for satin glass based on machine vision, Proc. IEEE I2MTC 2009, International
554 Instrumentation and Measurement Technology Conference, Singapore, pp. 288-293,
555 May 5-7 2009.
- 556 [39] F. Adamo, F. Attivissimo, A. Di Nisio and M. Savino, An automated visual inspection
557 system for the glass Industry, Proc. 16th IMEKO TC-4 International Symposium and
558 13th Workshop on ADC Modelling and Testing, Florence, Italy, pp. 442-447,
559 September 22-24 2008. ISBN 978-88-903149-3-3.
- 560 [40] Q. Qi, X. Jiang, X. Liu, P. J. Scott, An unambiguous expression method of the surface
561 texture, Measurement, vol. 43, pp. 1398-1403, 2010.
- 562 [41] S. Suzuki, K. Abe, Topological Structural Analysis of Digitized Binary Images by
563 Border Following, Computer Vision, Graphics and Image Processing, vol. 30, pp. 32-
564 46, December 1985.
- 565 [42] G. Bradski , The Open CV Librar, Dr. Dobb's Journal of Software Tools, 2000,
566 <http://www.drdobbs.com/open-source/the-opencv-library/184404319>
- 567 [43] R. Smith, The Tesseract Open Source OCR System,
568 <http://code.google.com/p/tesseract-ocr>.
- 569

Esame Richiesto	Risultato	U.M.	Valori di Riferimento
Sg-EMOCROMO			
WBC	9,04	$\times 10^3/\mu\text{L}$	4,00-10,00
RBC	4,63	$\times 10^6/\mu\text{L}$	4,00-5,50
HGB	13,3	g/dL	12,0-16,0
HCT	41,1	%	38,0-48,0
MCV	88,8	f	80,0-96,0
MCH	28,7	pg	27,0-34,0
MCHC	32,4	g/dL	32,0-37,0
PLT	286	$\times 10^3/\mu\text{L}$	140-500
RDW-SD	40,1	f	37,0-45,0
RDW-CV	12,5	%	11,0-14,0

Fig. 3. Expanded view of Fig. 2.

```

<?xml version="1.0" encoding="utf-8"?>
<medicalRecord version="2.0"><folder><labResults>
  <test>
    <nameRaw type="string">WBC</nameRaw>
    <nameIm array="[1 4]" type="numeric">[[ 247 1000 328 1036]]</nameIm>
    <name type="string">WBC</name>
    <resultRaw type="string">9,04</resultRaw>
    <resultIm array="[1 4]" type="numeric">[[1025 1000 1092 1036]]</resultIm>
    <result type="numeric">9.04</result>
    <unitRaw type="string">x1 O3/pL</unitRaw>
    <unitIm array="[1 4]" type="numeric">[[1412 1000 1556 1037]]</unitIm>
    <unit type="string">103/uL</unit>
    <testId type="numeric">17</testId>
  </test>
  ...
  <test>
    <nameRaw type="string">S-CREATININA</nameRaw>
    <nameIm array="[1 4]" type="numeric">[[ 185 2761 434 2800]]</nameIm>
    <name type="string">S-CREATININA</name>
    <resultRaw type="string">0,73</resultRaw>
    <resultIm array="[1 4]" type="numeric">[[1022 2761 1089 2800]]</resultIm>
    <result type="numeric">0.73</result>
    <unitRaw type="string">mg/dL</unitRaw>
    <unitIm array="[1 4]" type="numeric">[[1410 2761 1506 2800]]</unitIm>
    <unit type="string">mg/dL</unit>
    <testId type="numeric">32</testId>
  </test>
</labResults></folder></medicalRecord>

```

Fig. 4. An excerpt of an output XML file containing data extracted from medical laboratory tests of Fig. 1.

LIST OF TABLES

	Test name cell	Test result cell	Test measurement unit cell	Test reference range cell
Test row	Present	present	don't care	don't care
Result row	Absent	present	absent	absent
Name row	Present	absent	don't care	don't care

Table I. Row classification according to the cell types it contains

<i>test Id</i>	<i>name</i>	<i>unit</i>	<i>nameAliases</i>	<i>unitAliases</i>	<i>testHeaderId</i>	<i>header Type</i>	<i>multi Row</i>	<i>specialPa rsing</i>
1	RBC	10 ⁶ /uL	RBC	10 ⁶ /uL				
2	HGB	g/dL	HGB	g/dL				
10	RDW	%	RDW	%				
11	PLT	10 ³ /uL	PLT	10 ³ /uL				
21	LYMPH%	%	LYMPH%	%				
31	S-URATO	mg/dL	S-URATO	mg/dL				
44	CLEARANCE CREATININA	mL/min	CLEARANCE DELLA CREATININA, CLEARANCE CREATININA	mL/min,mL/ minute				
51	BILIRUBINA TOTALE	mg/dL	BILIRUBINA TOTALE	mg/dL				
105 2	BILIRUBINA FRAZIONATA		BILIRUBINA FRAZIONATA			2		
52	BILIRUBINA FRAZIONATA DIRETTA	mg/dL	BILIRUBINA FRAZIONATA DIRETTA, DIRETTA	mg/dL	0, 1052			
53	BILIRUBINA FRAZIONATA INDIRETTA	mg/dL	BILIRUBINA FRAZIONATA INDIRETTA, INDIRETTA	mg/dL	0, 1052			
106 2	CICLOSPORINA	ng/mL	CICLOSPORINA	ng/mL		1		
62	CICLOSPORINA T0	ng/mL	CICLOSPORINA T0, T0	ng/mL	0, 1062			
63	CICLOSPORINA T2	ng/mL	CICLOSPORINA T2, T2	ng/mL	0, 1062			
103	SODIO	mEq/L	SODIO	mEq/L, mmoli/L				
109	COLORE		COLORE					
120	PESO SPECIFICO	g/L	PESO SPECIFICO					1
121	ESAME MICROSCOPICO DEL SEDIMENTO		ESAME MICROSCOPICO DEL SEDIMENTO			1	1	

Table II. Knowledge base excerpt

Statistic name	Relative frequency	Absolute frequency	Description of the event counted by the statistic
<i>Test matches</i>	95.2 %	457	The <i>true test</i> matches an <i>estimated test</i> on the same page.
<i>Test mismatches</i>	4.8 %	23	The <i>true test</i> doesn't match any <i>estimated test</i> on the same page.
<i>Test names matches</i>	97.9 %	470	The <i>true test name</i> (or, equivalently, <i>testId</i>) matches an <i>estimated test name</i> on the same page.
<i>Test names mismatches</i>	2.1 %	10	The <i>true test name</i> (or, equivalently, <i>testId</i>) doesn't match any <i>estimated test name</i> on the same page.
<i>Numeric mismatches</i>	2.3 %	11	The <i>true test name</i> matches an <i>estimated test name</i> on the same page, but the <i>result</i> , which is expressed in numeric form, differs.
<i>String mismatches</i>	0.4 %	2	The <i>true test name</i> matches an <i>estimated test name</i> on the same page, but the <i>result</i> , which can't be expressed in numeric form, differs.
<i>Spurious test names</i>	0.0 %	0	The <i>estimated test name</i> (or, equivalently, <i>testId</i>) doesn't match any <i>true test name</i> on the same page.
<i>Aggregated errors</i>	4.8 %	23	Any event that contributes to <i>Test mismatches</i> or <i>Spurious test names</i> statistics.

Table III. Performance evaluation