# Document Structure Analysis Based on Layout and Textual Features

Stefan Klink, Andreas Dengel, Thomas Kieninger

German Research Center for Artificial Intelligence (DFKI, GmbH)
P.O. Box 2080, 67608 Kaiserslautern, Germany
E-mail: {klink, dengel, kieni}@dfki.de

**Abstract.** Document image processing is a crucial process in the office automation and begins from the 'OCR' phase with difficulty of the document 'analysis' and 'understanding'. This paper presents a hybrid and comprehensive approach to document structure analysis. Hybrid in the sense, that it makes use of layout (geometrical) as well as textual features of a given document. These features are the base for potential conditions which in turn are used to express fuzzy matched rules of an underlying rule base.

## 1 Introduction

In the office automation context processing, filing and retrieving of documents are the essential functionalities that help to increase the productivity of our work. Within this application domain, document analysis and understanding are the crucial activities for integrated office automation systems. However, despite major advances in computer technology, the degree of automation in acquiring and inquiring data from documents is still very limited. Most of the existing document analysis systems are restricted to relatively small application domains (e.g. invoices, cheque amounts, newsletters). Even if some of the systems can be adapted to a new domain, this adaptation would be as time consuming as developing a new system from scratch. Therefore, we investigated an approach, which could be adapted easily to arbitrary application domains and thereby provides a high degree of automation.

One of the aspects is how to keep the human operations as simple and reasonable as possible. The maximum use of document structure (e.g. layout structure and logical structure) **[2]** in analyzing documents and the exchangeable rule base allows the system to be easier adapted to a new domain.

## 2 Related Work

The analysis of document logic has not received as much attention as geometrical layout analysis, but however, some interesting methods of page understanding have been proposed **[3]**-**[13]**. The greatest part of them are rule-based techniques, where two big classes of methods may be identified: methods based on Tree Transformations **[13]** and methods based on Formatting Knowledge **[8][9]**.

Techniques belonging to the first class try to modify the geometrical layout tree moving nodes within the tree and annotating each node with the appropriate logical label, according to specific sets of rules. An example of such a method is the Multi-Articled Understanding **[13]**.

A different approach is based on the preliminary knowledge of the layout of the page, in order to optimize the logic rules that correlate with the document features. Examples of Formatting Knowledge methods are the characterization of the most relevant blocks in a page **[6]** or the application of a syntactic grammar **[9]**. A recent paper **[10]** is directed to our task, but uses different techniques.

Our system is not only a new comprehensive combination of several approaches but it integrates new strategies and ideas. First of all, we don't only use geometrical information as in **[3]**, but additionally we make use of textual information (e.g. text pattern), font information (e.g. style, size, boldness, etc.) and universal attributes (e.g. number of lines) given by the OCR. The most important extension to other approaches is the ability to express relations that can be determined between different layout objects (e.g. an author is (directly) placed under a title).

## 3 Document Model (Layout and Logical Structures)

**Fig. 1** visualizes our document model. It mainly consists of two parts: the physical (left) and the logical (right) part:

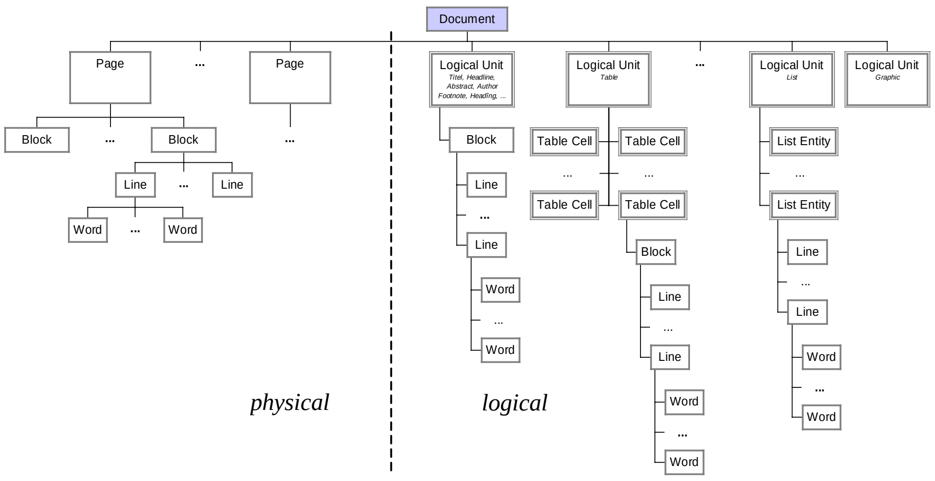


**Fig. 1:** Physical and logical part of the document model

From the physical point of view, the document has some pages. A page consists of some blocks, a block consists of some lines, and a line consists of some words. The document must contain at least one object of each type.

From the logical point of view, the document contains *logical units*, e.g. a heading, a title, tables or lists etc. A logical unit has several characteristic appearances: it can either be a title, heading, abstract, author, footnote, headline etc. In these cases the logical units consist of exactly one block. These logical units can be assigned with various labels each of them carrying a probability value (e.g. 80% heading, 20% title).

But logical units could also be aggregations of blocks to either a list or a table. Each of these two types will have exactly one label (100% list or table respectively). While a list consists of a (one-dimensional) sequence of blocks, which themselves are labeled as list items (or sub-lists), tables are even more complex: they are a (two-dimensional) array-like structure with even some irregularities.

## 4 Document Understanding

The recognition of a document is split up into several phases. First, the document is scanned. Then, it is segmented and the characters and their fonts are recognized by a commercial OCR system. We use Textbridge, wherein these information are stored in the XDOC-format. This file including its text, font and layout attributes can be used to conduct further analysis on the text and to find the "structured" parts of the document **[5]-[11]**. The XDOC-file is converted into our proprietary GT-format, which represents the input of our system. The results are added to the document tree and are stored within the GT-file for further analysis and retrieval steps.

The recognition of the document structure is divided into two different phases of processing: first, all common document structures like headings, footnotes and lists are recognized and second, all domain dependent logical elements are recognized using the appropriate rule base. For the first phase, more granularity is needed to express enumerated list entities within a list, table cells within tables or to aggregate lines to headings or footers.

For the second phase, all rules of the rule base will be checked independently against all blocks using geometric, font specific and textual features. If a rule fires, the block will get the appropriate label. Blocks will neither be resegmented nor will some sub-objects be rearranged.

### 4.1 Common document structure recognition

In this processing phase, common structures like header, footer or lists are recognized independently.

The determination of a document's header is not very sophisticated and needs some more attention. But in the current release, the system looks at the top of the document and searches a line which is not 'too far away'[1] from the top (in other words, which has a small top margin). If a line is found, it will establish the document header. Then, the system sequentially adds all lines downwards to the document header until a line has a 'too big'[2] distance to the growing header. In the case, that the document's header is reaching more than a certain amount of the vertical dimension of the page (e.g. a quarter), the system rejects the header assumption.
The recognition of the footer is similar.

[1] The decision is driven by a given threshold.

For recognizing lists, several more or less successful strategies are published in literature **[14][15][16]**. Due to our domains, we just need to recognize flat lists which are not recursive. Assuming that the segmentation of the OCR-System on the word level is correct, our system is able to recognize enumerated, bulleted and dashed lists.

### 4.2 Domain dependent logical labeling

The second phase is the rule based approach. Surely, this is the most complex phase, due to the variety of possible cases and different dispositions within a page. That is why we did not predefine any labels or restrict them in any way. Furthermore, the rule base is exchangeable and editable during the runtime of the system, that means it is adaptable to the document domain, e.g. business letters, journals, etc. For each domain, an appropriate rule base could be created and successively be tuned to it.

For the logical labeling, each (physical) block is investigated in isolation and the rule interpreter tests are applied, whether one of the rules is firing. In this case, the block is assigned with the appropriate logical label. With this approach, a block might get more than just one label, e.g. '80% heading, 20% title'. It depends on how many and which rules have fired upon that block.

To rate alternate labels, we use a probabilistic approach. The ratings of each label depend on the weight of the defined rule attributes (see chapter 5) and on the confidence of match of each attribute. The more and 'better' the attributes are fulfilled the higher is the rating of the label.

Consequently, each attribute match is a fuzzy match. E.g. a block is not only 'bold' or 'not bold' nor is it only 'within a specified region' but when most of the words are bold and some of them are regular, then a rule with the condition 'has to be bold' will be matched with maybe 90%. Or if a block is not exactly inside a specified region, the condition 'has to be within the region xy' will match with some rate. Each fuzzy values of all defined attributes are aggregated with a mean function to a final rating.

## 5 Rule structure

For each label, exactly one rule is defined. Due to the complexity of the labeling task, the structure of a rule is non-trivial. The rules are only used to find domain dependent objects. The recognition of the common document structures (headings, lists, etc.) is done in the previous phase with hard-coded algorithms.

### 5.1 Rule expression

A rule in our system is not only a simple if-then rule but rather consists of a logical expression which contains the so called *rule units* combined with the logical operations 'and', 'or', 'not' and parentheses, e.g. $( A * B ) + ( C * \neg D )$.

Testing, if a rule is firing (for a block) or matching the rule against the block respectively means evaluating this logical expression. The logical OR (+) is evaluated by a maximum function, the logical AND (*) is evaluated by an average function, and the logical NOT ($\neg$) is calculated as complement to 1, e.g. $\neg$ D is evaluated as (1-D).

### 5.1.1 Rule units

While evaluating the logical expression, each rule unit (e.g. A) is matched against the actually investigated block. The match method returns a normalized value between 0.0 and 1.0. This value represents the confidence with which the rule unit is matching.

A rule unit is divided into two parts which themselves define self-related and a list of cross-related attributes to express relations between the actually inspected block and several other blocks.

### 5.1.2 Self-related attributes

The self-related attributes define only the features of the actually inspected block. These might consist of morphological features of the physical components, such as dimensions, position in the page, number and alignment of lines and font attributes defined such as font family, size, boldness, etc. Additionally, a list of containing strings, e.g. ‘with best regards’ and a regular expression, e.g. ‘[0-3]*[0-9].[0-1]*[0-9].[0-9] [0-9]’ or word patterns, e.g. ‘dear mr?’ could be defined.

With these attributes, rules could be defined such as ‘an address must be within a certain position, must have more than two lines and must contain a zip-code’.

### 5.1.3 Cross-related attributes

The cross-related attributes could be divided into three main parts:

*1. Geometrical relations:*

The first part expresses geometrical relations between the actually inspected block and another one, e.g. “Another block (possibly with specific attributes or labels) must be located within a qualified region”. These regions can either be qualified relative to the observed object or through document coordinates.

The relative positions are defined using a two-dimensional extension of Allen’s time-relations **[1][3]**. Thus it is possible to express “block A must be underneath B“.

During our extensive experiments, we realized that these qualitative specifications with Allen's relations lack the ability to quantify distances. Thus it would not be possible to express that e.g. “a caption has to be located underneath a table” as this “underneath” can apply to any object regardless of its distance.

Hence, we extended the qualitative aspect with ranges of distance measures. It is possible to extent the above example to something like: “a caption has to be located underneath a table; the distance should be more than 1 and should not exceed 4 times the line height of the caption”.

*2. Textual relations:*

The second part specifies the absolute or relative number of common words within both blocks or a given list of predefined words that need to be located in the related block. This allows the definition of rules like: “There must be a block which has a least one word in common with the block for which the rule might be applied and furthermore it needs to contain the words ‘Dear’ and ‘Mr.’ ”. This could for instance express a relation between the recipient field of a letter and the salutation field.

*3. Label relation:*

Finally, a cross related block can be qualified through a specific label. The label condition is very usefully particularly with regard to document objects with are depending on each other. This attribute induces a very interesting effect:

The occurrence of one logical object triggers the labeling of other objects ! E.g. if the ‘title’ is found on a scientific paper, then the ‘author’ (or an ‘abstract’) is usually found underneath the ‘title’. Thus, if no ‘title’ exists on the document, no ‘author’ (or ‘abstract’) will exist too and the precision of an object as e.g. ‘author’ will increase.

As a matter of course, all these parts could be combined to rules like ‘There must be another block with the label ‘sender’ which is 300 up to 1800 pixels above and has at least one word in common.’

### 5.1.4 Match of each attribute

As mentioned above, each attribute is matched using the probabilistic approach of a fuzzy-match. E.g. a block is not only ‘within a certain region’ if it is completely located inside this region but even when its area overlaps this region partially. And two blocks have not only some words in common but it is absolutely possible that the word in the compared block is just similar.

For each self-related attribute a special match function $\mu^s$ and for each cross-related attribute a match function $\mu^c$ is defined which compares the defined rule attribute with the appropriate value of the actually inspected block and returns a probability value. In the following, these two types of match functions are introduced:

Let $b$ be the actually inspected block of the document,
$b_c$ be a block which is cross-related to $b$ and
$r$ be the actually inspected rule.
Let the self-related attributes be enumerated with $i$ and the cross-related attributes with $j$.
Let $v_i^b$ be the value of the block $b$ of attribute $i$ and
$\underline{v}_i^r$ and $\overline{v}_i^r$ be the value range of the attribute $i$ defined in the rule $r$ with $[\underline{v}_i^r, \overline{v}_i^r]$.

Each ***match function*** $\mu_i^s \in [0, 1]$ is defined as a mapping to a positive floating point number with:

$$\mu_i^s(v_i^b, \underline{v}_i^r, \overline{v}_i^r) = \begin{cases} 1.0, \text{if } \delta_i(v_i^b, \underline{v}_i^r, \overline{v}_i^r) = 0 \\ \beta_i(v_i^b, \underline{v}_i^r, \overline{v}_i^r, \delta_i(v_i^b, \underline{v}_i^r, \overline{v}_i^r)), \text{if } \delta_i(v_i^b, \underline{v}_i^r, \overline{v}_i^r) \le \varepsilon_i \\ 0.0, \text{else} \end{cases} \quad \textbf{(1)}$$

The match function returns 1.0 (which indicates a 100% match or a zero distance respectively) if the attribute value $v_i^b$ is perfectly within the appropriate rule value range $[\underline{v}_i^r, \overline{v}_i^r]$. To identify this and to identify if the attributes value is not within this range and additionally in this case, to calculate the distance between the attribute value $v_i^b$ and the rule defined value range $[\underline{v}_i^r, \overline{v}_i^r]$ we use for each attribute a particular distance function $\delta_i$.

Each ***distance function*** $\delta_i$ is individual because the distance for the attribute ‘number of lines’ must be calculated in a different way than for the attribute ‘font family’ or ‘contains string’ (e.g. ‘Dear Mr.’) or for regular expressions respectively. Basically, we use three different types:

A ***distance function for numerical values*** is just a plain substraction. For two dimensional attributes like dimensions or position (region) coordinates the distance function is similar. This distance function is used for attributes like ‘number of lines’, ‘font size’, etc:

$$\delta_i^s(v_i^b, \underline{v}_i^r, \overline{v}_i^r) = \begin{cases} \underline{v}_i^r - v_i^b, \text{if } v_i^b < \underline{v}_i^r \\ v_i^b - \overline{v}_i^r, \text{if } v_i^b > \overline{v}_i^r \\ 0.0, \text{else } (\text{ if } v_i^b \in [\underline{v}_i^r, \overline{v}_i^r]) \end{cases} \quad \textbf{(2)}$$

A ***distance function for strings*** is implemented with the Levenshtein string distance **[17]** which calculates the edit distance of two strings (here: $v_i^b$ and $v_i^r$). In the case of strings no value range but just discrete strings are defined in the rule.

A ***distance function for discrete classes***, e.g. ‘font family’ is defined with:

$$\delta_i^s(v_i^b, \underline{v}_i^r, \overline{v}_i^r) = \begin{cases} 0.0, \text{if } v_i^b = v_i^r \\ 2 * \varepsilon_i, \text{else} \end{cases} \quad \textbf{(3)}$$

Similar to the string type, no value range can be defined in the rule (here: $\underline{v}_i^r := \overline{v}_i^r$).

If the distance $\delta_i$ is zero or greater than ε the match function is nothing special. But if the distance is lower or equal than ε the fuzzy value which is within [0.0, 1.0] must be calculated. To do this, a ***boundary function*** $\beta_i$ is established individually for each attribute. The boundary function is defined as a linear function, a polynomial function or the well-known Sigmoid function. It depends on the attribute which function is used. If the values are varying very much, e.g. coordinates are varying between 1 and 3000, a linear or a Sigmoid function is used. Otherwise a polynomial function is used to get a steeper descent within small distances, e.g. font sizes are varying just a few pixels.

Each ***match function*** $\mu_j^c \in [0, 1]$ is defined as a mapping to a positive floating point number with:

$$\mu_j^c(v_j^b, v_j^{b_c}, \underline{v}_j^r, \overline{v}_j^r) = \begin{cases} 1.0, \text{if } \delta_j(v_j^b, v_j^{b_c}, \underline{v}_j^r, \overline{v}_j^r) = 0 \\ \beta_j(v_j^b, v_j^{b_c}, \underline{v}_j^r, \overline{v}_j^r, \delta_j(v_j^b, v_j^{b_c}, \underline{v}_j^r, \overline{v}_j^r)), \text{if } \delta_j(v_j^b, v_j^{b_c}, \underline{v}_j^r, \overline{v}_j^r) \leq \varepsilon_j \\ 0.0, \text{else} \end{cases} \quad \textbf{(4)}$$

Again, the ***distance functions*** $\delta_j^c$ are individual for each cross-related attribute. They are absolutely similar to the distance functions $\delta_i^s$ of the self-related attributes and it is not necessary to write them down explicitly. The only difference to them is that not the values itself are compared to the values (value ranges) defined in the rule-unit but now the relation of the values of the block $b$ and the values of the related block $b_c$ are compared with the appropriate relation defined in the rule.

### 5.1.5 Assembling each attribute match value

All the match values of each self-related and cross-related attribute must be combined to a final match value which indicates the probability for this label. This is done by a weighted mean function:

Let $b$ be the actually inspected block of the document,
$b_n$ be a cross-related block to $b$,
$n_c$ be the number of cross-related blocks and
$l$ be the label defined by a rule $r$.

The ***probability*** $\Pi \in [0, 1]$ of labeling a block $b$ with a label $l$ is defined as a mapping of all blocks and all labels to a positive floating point number with:

$$\Pi(b, b_1, \ldots, b_{n_c}, r) = \frac{1}{1 + n_c} * \left( \sum_i \omega_i * \mu_i^s (v_i^b, \underline{v}_i^r, \overline{v}_i^r) + \sum_j \sum_{n=1}^{n_c} \omega_j * \mu_j^c (v_j^b, v_j^{b_n}, \underline{v}_j^r, \overline{v}_j^r) \right) \tag{5}$$

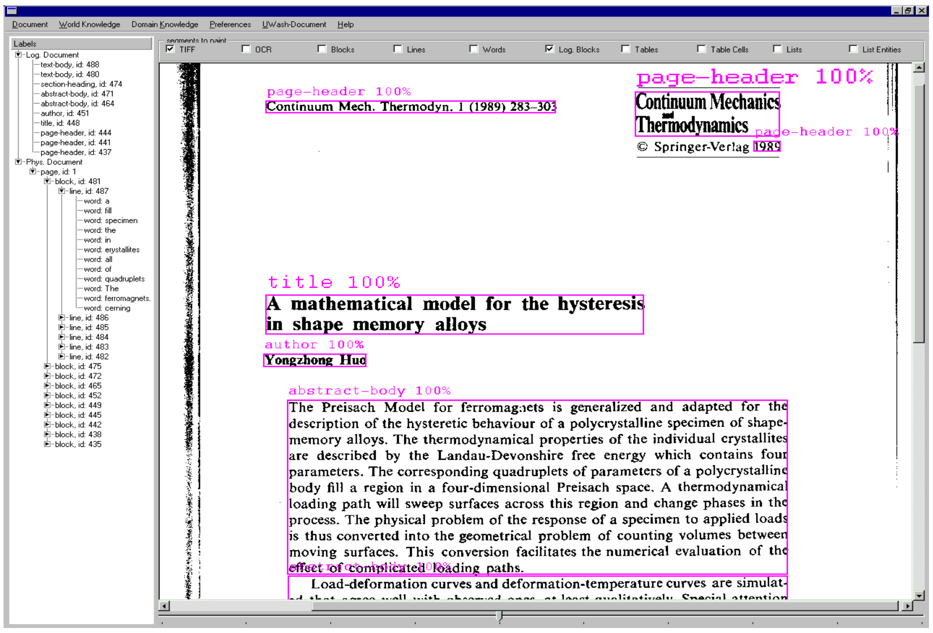
Each match function is weighted with a factor $\omega \in [0, 1]$ which can be defined individually for each attribute within the rule.

## 6 Experimental Results

Our approach has been implemented in a prototype and has been successfully tested on two document domains, business letters and technical journal pages.

### 6.1 Evaluation and visualization front-end

To evaluate the results, the following visualization system is implemented (**Fig. 2**):



[✓ TIFF] OCR Blocks Lines Words [✓ Log. Blocks] Tables Table Cells Lists List Entities

- Log. Document
  - text-body, id: 488
  - text-body, id: 480
  - section-heading, id: 474
  - abstract-body, id: 471
  - abstract-body, id: 464
  - author, id: 451
  - title, id: 448
  - page-header, id: 444
  - page-header, id: 441
  - page-header, id: 437
- Phys. Document
  - page, id: 1
    - block, id: 481
      - line, id: 487
        - word: a
        - word: fill
        - word: specimen
        - word: the
        - word: in
        - word: erystallites
        - word: all
        - word: of
        - word: quadruplets
        - word: The
        - word: ferromagnets.
        - word: cerning
      - line, id: 486
      - line, id: 485
      - line, id: 484
      - line, id: 483
      - line, id: 482
    - block, id: 475
    - block, id: 472
    - block, id: 465
    - block, id: 452
    - block, id: 449
    - block, id: 445
    - block, id: 442
    - block, id: 438
    - block, id: 435

page-header 100%
Continuum Mech. Thermodyn. 1 (1989) 283–303

page-header 100%
Continuum Mechanics and Thermodynamics

page-header 100%
© Springer-Verlag 1989

title 100%
A mathematical model for the hysteresis in shape memory alloys

author 100%
Yongzhong Huo

abstract-body 100%
The Preisach Model for ferromagnets is generalized and adapted for the description of the hysteretic behaviour of a polycrystalline specimen of shape-memory alloys. The thermodynamical properties of the individual crystallites are described by the Landau-Devonshire free energy which contains four parameters. The corresponding quadruplets of parameters of a polycrystalline body fill a region in a four-dimensional Preisach space. A thermodynamical loading path will sweep surfaces across this region and change phases in the process. The physical problem of the response of a specimen to applied loads is thus converted into the geometrical problem of counting volumes between moving surfaces. This conversion facilitates the numerical evaluation of the effect of complicated loading paths.

abstract-body 100%
Load-deformation curves and deformation-temperature curves are simulat-

**Fig. 2.** Our visualization system showing a journal page with labels

The visualization front-end displays the document image and the layout objects (segments) using their bounding box. In addition, the assigned labels (if any) together with their probabilities are displayed at the top of each bounding box.

If an object is labeled wrongly then the label could be edited by selecting the object or the front-end assists the user in modifying the existing rules or creating new ones to mark the object with the right label with a higher probability.

An additionally helpful feature is the automatically rule generation. After selecting all related objects and giving a label name, all self-related and cross-related attributes are automatically ascertained. The system saves all fulfilled conditions as rule units and a conjunction of the rule units as the default logic using the proper rule syntax. An administrator will now be able to delete unimportant units, change the logic and assign individual weights. It hence helps preventing syntactically incorrect rule definitions and also provides the complete list of generally available features.

## 6.2 Test documents

### 6.2.1 Incoming business letters

Our test documents are 89 business letters with a length of one or two pages drawn from incoming letters to our company. The documents are sufficiently structured to get a non-trivial test.

They contain headings, footnotes, lists (enumerated and bulleted), addresses, subjects, dates, sender, closure, etc. See **Fig. 3** for some sample thumbnails:

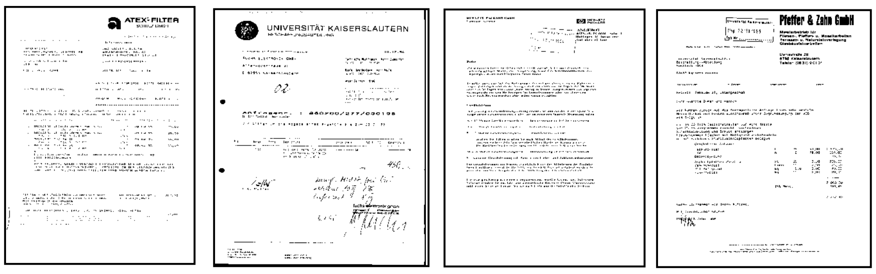


**Fig. 3.** Sample thumbnails of typical business letters

### 6.2.2 U-Wash document corpus

Another test set is the technical document database of the University of Washington **[18]** with 979 journal pages over a large variety of journals from diverse subject areas and publishers. Each page has zone ground-truth represented by almost non-overlapping rectangles and labeling ground-truth for each logical block. The character recognition and the word and line segmentation is done by the OCR-System Text-bridge (see chapter 4) but the segmentation of the blocks is synchronized with the U-Wash ground-truth zoning to get the best possible 1:1 correlation of the labeling data.

We adapted one single rule-base with one rule for each label and elaborated the recall and the precision over all documents. Goal of this test was to establish one single rule-base for the complete document corpus and not to use document classification with one rule-base for each class. We define rules for the labels: *abstract-body, abstract-heading, affiliation, biography, caption, drop-cap, highlight, keyword-body, keyword-heading, keyword_heading_and_body, membership, page-number, pseudo-code, publication-info, reader-service, synopsis,* and *text-body*.

## 6.3 Results

### 6.3.1 Incoming business letters

Our prototype has no labeling problems with incoming business letters. For almost all documents the correct logical structure and the domain specific layout primitives (e.g. subject, date, etc.) are found. The main reasons for the good results are the clear distinction of the defined layout primitives and the structured form of the layout objects. Thus, the (man made) rules could be defined easily and could be adapt to all documents with great effort.

Errors occur only if the segmentation fails or the OCR recognition supplies wrong text or wrong attributes respectively. Then the rules that are based on text pattern, e.g. date or subject, could not find the right layout objects accurately.

### 6.3.2 U-Wash document corpus

The journal documents are more tricky and during the document analysis process several problems arised:

First of all, the OCR problem should be mentioned. As said above, the block segmentation of Textbridge is not usable at all. To get an accurate segmentation we need to resegment all blocks beginning from the word level directly after loading the document.

The second OCR problem is the partly insufficient character recognition. The OCR is not capable of handling (parts of) documents with an angle of about 90 degrees. Here, the results are absolutely unusable. Additionally, on a lower image quality, the OCR results of bold fonts and bad copies go worse. This interferes particularly those labels which are based on text pattern, e.g. abstract, caption or keyword

Another serious problem is the similarity of some logical blocks. Some labels are often mixed up and the precision is less than expected. Typical examples for this problem are the labels footnote, abstract, text-body and biography.

But nevertheless, our rule based approach with the fuzzy matching achieves considerable results. If we apply the rule base with all defined labeling rules over all documents the system obtains the following recall and precision results (see **Fig. 4**):

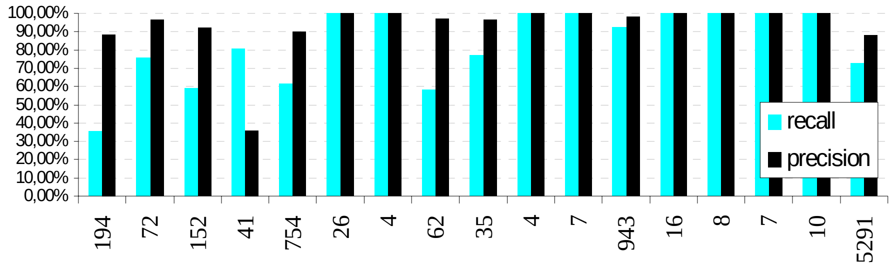


**Fig. 4.** Recall and precision of each defined label within the rule-base

The y-axis shows the value of the recall or precision respectively and the x-axis shows the number of occurrence of each defined label.

The precision of 100% for some labels can be explained on the one hand with good OCR (character & segmentation) results and a low similarity to other logical objects.

Another reason is, that for some of these labels just a few logical objects exist. As a matter of course, the rule can be well adapted for these objects.

For example for the label ‘membership’ the text pattern ‘member’ and ‘fellow’ is sufficient for most of the 7 ‘membership’ objects. Even for the 943 ‘page-number’ objects a plain enumeration criterion with some additional relationship attributes is enough to find over 90% of them with a 98% precision. For better recalls, the rule must be more specific and must cover various special cases.

But for all these results it should be mention that up to now all rules are man-made and that the recall could moreover improved with some further rule-unit extensions. Especially, the rules for the 194 ‘abstract’ objects and the 41 ‘biography’ objects have still some potential for improvements because up to now not all characteristics and potentialities of the rules are exhausted.

A further approach, which will help to reduce the complexity of the rules will be to integrate a document classifier with an automatically selection of the appropriate rule-base. Then, not only one single rule must find all variations of e.g. an ‘abstract’ but for each document class a special rule for each logical object could be defined within each appropriate rule-base. These rules will be more exact, more precise and less complex. Especially the rules with lower precision will profit from this.

## 7 Conclusions and Outlook

In this paper we have presented a rule-based approach for document structure recognition with an exchangeable rule base adaptable to several domains. The rules contain textual and geometrical layout features. They not only use self-related attributes but additionally make use of cross-related attributes to express relationships between several blocks. Furthermore, the rule units are combined with any logical expression to an if-then rule with a complex condition part. These features give enough power to express all logical objects within our tested domains.

The complexity of the rules is a powerful feature but at the same time its greatest shortcoming. Up to now, the creation, testing and adaptation of rules is man-made and a very time consuming task. For doing this, the visualization tool helps the user to define a rule automatically by a positive label example with some mouse clicks. This rule could be easily changed and adapted (‘learned’) to other label examples during the run-time of the labeling prototype.

Our next step will be to implement a rule learning capability in the way that the rules could be learned from a set of ground-truth documents **[19]**. For this task, several strategies are prepared and a comparison and/or voting will show the best one.

The fuzzy matching results enable to draw conclusions from the interference between some rules. Mix-ups of groups of logical objects clearly show similarities which could be expressed qualitative to see ‘how’ similar e.g. an ‘abstract-body’ is to a ‘text-body’. To do this, special criteria and similarity measures must be worked out.

Due to the fact that a suitable document representation and retrieval in the WWW is getting more and more important and that all needed information are assembled during the document analysis and understanding steps, the idea to provide a document output in (HTML)/XML format is self-evident. Thus, a direct transformation from the scanned document image into an appropriate presentation in the Web will be possible

**[20][21]**. The main goal should be to give an exact 1:1 presentation including all structural and layout (font sizes, distances, etc.) features used in the document image.

## References

[1] James F. Allen: "Towards a General Theory of Action and Time", in Artificial Intelligence 23 (1984) 123-154

[2] ISO 8613: "Information Processing-Text and Office Systems-Office Document Architecture (ODA) and Interchange", International Organization for Standardization, 1989

[3] Hanno Walischewski: "Automatic Knowledge Acquisition for Spatial Document Interpretation", in Proc. of the 4th Intern. Conference on Document Analysis and Recognition (ICDAR), Aug 1997, pp. 243-247

[4] Thomas A. Bayer and Hanno Walischewski: "Experiments on extracting structural information from paper documents using syntactic pattern analysis", in Proceedings of the 3rd ICDAR, Aug 1995, pp. 476 - 479

[5] Xuhong Li, Peter A. Ng: "A Document Classification and Extraction System with Learning Ability", in Proceedings of the 5th ICDAR, Sep 1999, pp. 197 - 200

[6] Antonacopoulos and A. Coenen: "Region Description and Comparative Analysis Using a Tesseral Representation", in Proceedings of the 5th ICDAR, Sep 1999, pp. 193 - 196

[7] Yuan Y. Tang, Chang D. Yan, Ching Y. Suen: "Document Processing for Automatic Knowledge Acquisition", in IEEE Trans. on Knowledge and Data Engineering, Vol. 6, No.1, Feb 1994, pp.3-21

[8] Dengel: "Document Image Analysis - Expectation-Driven Text Recognition", in Proceedings of the Conf. on Syntactic and Structural Pattern Recognition, 1990, pp. 78-87

[9] M. Viswanathan: "Analysis of Scanned Document - A Syntactic Approach", in Proceedings of the Conf. on Syntactic and Structural Pattern Recognition, 1990, pp. 450 - 459

[10] O. Altamura, F. Esposito, D. Malerba: "WISDOM++: An Interactive and Adaptive Document Analysis System", in Proceedings of the 5th ICDAR, Sep 1999, pp. 366 - 369

[11] Robert M. Haralick, "Document Image Understanding: Geometric and Logical Layout" in Proc. of the Conf on Computer Vision and Pattern Recognition, Washington Seattle, 1994

[12] Tsukasa Kochi, Takashi Saitoh: "User-defined Template for Identifying Document Type and Extracting Information from Documents" in Proceedings of the 5th ICDAR, Sep 1999, pp. 127 - 130

[13] S. Tsujimoto, H. Asada: "Understanding Multi-articled Documents", in Proc. of the 10th International Conference on Pattern Recognition, Atlantic City, N.J., 1990, pp. 551 - 556

[14] S. Golub: "txt2html - Text to HTML converter", http://www.aigeek.com/txt2html/

[15] K. Summers: "Automatic Discovery of Logical Document Structure", PhD thesis, Cornell University, Aug 1998

[16] R. Brugger: "Eine statistische Methode zur Erkennung von Dokumentstrukturen", PhD thesis, University of Freiburg, Mai 1998

[17] V. I. Levenshtein: "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", in Soviet Physics Doklady, Vol. 10, Nr. 8, Feb 1966, pp. 707 - 710

[18] T. Phillips: "User's Reference Manual", UW-I English/Technical Document Image Database, University of Washington, Aug 1993

[19] M. Junker, A. Abecker: Learning Complex Patterns for Document Categorization", AAAI/ICML Workshop on Learning for Document Categorization, 1998

[20] S. J. Simske: "The Use of XML and XML-Data to Provide Document Understanding at the Physical, Logical and Presentational Levels", Proc. of the International Workshop on Document Layout Interpretation and its Applications (DLIA), Sep 1999

[21] Y. Wang, T. Philips and R. Haralick: "From Image to SGML/XML Representation: One Method", Proc. of the DLIA, Sep 1999