# Improved CHAID Algorithm for Document Structure Modelling

Abdel Belaïd, Philippe Moinel, Yves Rangoni

## To cite this version:

Abdel Belaïd, Philippe Moinel, Yves Rangoni. Improved CHAID Algorithm for Document Structure Modelling. Document Recognition and Retrieval XVII, Jan 2010, San Jose, United States. pp.7, 10.1117/12.839794 . inria-00579684

HAL Id: inria-00579684
https://hal.inria.fr/inria-00579684

Submitted on 24 Mar 2011

# Improved CHAID Algorithm for Document Structure Modelling

A. Belaïd, T. Moinel, Y. Rangoni

LORIA-University Nancy 2, Campus Scientifique, B.P. 239, Vandœuvre-Lès-Nancy, France

## ABSTRACT

This paper proposes a technique for the logical labelling of document images. It makes use of a decision-tree based approach to learn and then recognise the logical elements of a page. A state-of-the-art OCR gives the physical features needed by the system. Each block of text is extracted during the layout analysis and raw physical features are collected and stored in the ALTO format. The data-mining method employed here is the "Improved CHi-squared Automatic Interaction Detection" (I-CHAID). The contribution of this work is the insertion of logical rules extracted from the logical layout knowledge to support the decision tree. Two setups have been tested; the first uses one tree per logical element, the second one uses a single tree for all the logical elements we want to recognise. The main system, implemented in Java, coordinates the third-party tools (Omnipage for the OCR part, and SIPINA for the I-CHAID algorithm) using XML and XSL transforms. It was tested on around 1000 documents belonging to the ICPR'04 and ICPR'08 conference proceedings, representing about 16,000 blocks. The final error rate for determining the logical labels (among 9 different ones) is less than 6%.

**Keywords:** Document Image Analysis and Recognition, Physical and logical layout analysis, OCR, Improved CHAID Algorithm, XML based formats

## 1. INTRODUCTION

Reverse engineering is becoming a common tool for document structure recognition of raw images capable of reaching good recognition results.[12] This task involves an implicit document structure modelling, whose construction is made difficult due to the complexity of the mapping between physical and logical structures. Both structures are rarely unique, and in most of the cases, there is no unambiguous mapping between them.

Some works proposed automatic procedures for modelling[1,3] or formats as DAFS[6] have been developed for this purpose, but, it seems that none of them have been generally adopted, indicating that they may not be extensible and general enough to comply with the requirements.

In this paper, we assume that the physical layout analysis is performed by a state-of-the-art OCR, and we propose a generic data-mining approach to tackle the problem of the logical structure recognition. The kernel of the system relies on a decision tree: the "Improved CHi-squared Automatic Interaction Detection" (I-CHAID) algorithm. It uses results of the OCR converted in ALTO format to create a model of the physical layout. On top of the syntactic rules generated from the document description, we add structural and logical rules to support the original tree for finding the final logical structure of the document.

## 2. IMPROVED CHAID ALGORITHM

The Improved CHAID Algorithm,[9] is based on data discrimination by trees. Decision tree is one common method used in data mining to extract predicted information. The pioneers of this work were Morgan and Sonquist.[11] They used the regression trees in a prediction and explanation process: AID (Automatic Interaction Detection). Several discrimination and classification methods followed, based on the same representation paradigm by trees: THAID (Theta AID) and CHAID (CHi-squared Automatic Interaction Detection).[9] In machine learning, most of the techniques are based on information theory. The first method was ID3 proposed by Quinlan (Induction of decision tree).[14] Many heuristics were proposed in Nineteenth Century to improve the behaviour of the Quinlan's system, leading to the famous C4.5 method. This mobility emerged the concept of lattice graphs[16] which was popularised by the induction graphs of the SIPINA method.[5]

In his SIPINA software, Rokotamalala exposes the principle of the decision tree construction for classification and discrimination problems. The problem is to predict the output value (or the class) of an object from a set of variables, discretes or continuous. It is a question of finding a partitioning of the individuals that we can represent by a decision tree. The objective is to produce individual groups, the most homogeneous as possible from the point of view of the variable to be predicted. The idea is to represent the empirical distribution of the attribute to be predicted by each node of the tree. Thus, the tree build favours the more "discriminating" attributes.

Here, the difficulty is to choose among $N$ attributes characterising the structure elements that made it possible to have the best discrimination rate. There is a great number of information or statistical criteria, the most used is the entropy of Shannon and its alternatives. Another way of characterising the segmentation is to measure the causality between the variable candidate and the variable to be predicted.

For the process comprehension, it should be noted that a segmentation makes it possible to define a contingency table crossing the variable to be predicted and the descriptor candidate. In the following, we will adopt the notations to describe the numbers resulting from the crossing of the attribute class with $K$ modalities and a descriptor with $L$ methods.

Table 1. Number table during the crossing of two variables

| $Y/X$ | $x_1$ | $x_l$ | $x_L$ | $\Sigma$ |
|---|---|---|---|---|
| $y_1$ | | | | |
| | | . | | |
| | | . | | |
| | | . | | |
| $y_k$ | ... | $n_{kl}$ | ... | $n_l$ |
| | | . | | |
| | | . | | |
| | | . | | |
| $y_K$ | | | | |
| $\Sigma$ | | $n_k$ | | $n$ |

To evaluate the relevance of a variable in the segmentation, CHAID proposes the independence deviation $\chi^2$ defined by Eqn. 1.

$$\chi^2 = \Sigma_{k=1}^{K} \Sigma_{l=1}^{L} \frac{(n_{kl} - \frac{n_k \times n_l}{n})^2}{\frac{n_k \times n_l}{n}} \tag{1}$$

The values of $\chi^2$ are not bounded, they are in the range $[0, +\infty[$. The main drawback is the high emphasis of the descriptors having a high number of modalities. To reduce this negative impact, it is much more suitable to normalise by the number of freedom degrees. The formula $T$ of Tschuprow (Eqn. 2) has values now in a range $[0, 1]$. This new equation gives the Improved CHAID algorithm.

$$T = \frac{\chi^2}{n\sqrt{(K-1) \times (L-1)}} \tag{2}$$

.

The I-CHAID used in this paper has the following parameters:

- "P-level" for splitting nodes: The computed p-value of the $\chi^2$ statistic is compared to this threshold. When it is greater, the split process is not performed. If we decrease the "P-level" threshold, we obtain a shorter tree.

- "P-level" for merging nodes: The merging process compares the class distribution in the leaves. It searches for the two most similar leaves. They are merged together if the difference between distributions is not significant enough. The statistical significance is determined by comparing the computed P-value of the test with this "P-level" threshold defined by the users. If we decrease the threshold, we obtain larger trees. If we increase the threshold, we obtain a more compact tree.

- Bonferroni adjustment: as we are multiplying the tests at each node to find the best solutions, we increase in the same time the probability of finding a good solution. The true "P-value" of these tests must be suitable with this kind of adjustment.

## 3. APPLICATION TO DOCUMENT ZONE LABELLING

Based on the Improved CHAID algortihm for the document model generation, we propose a flexible and generic approach to deal with several document classes (Fig. 1)

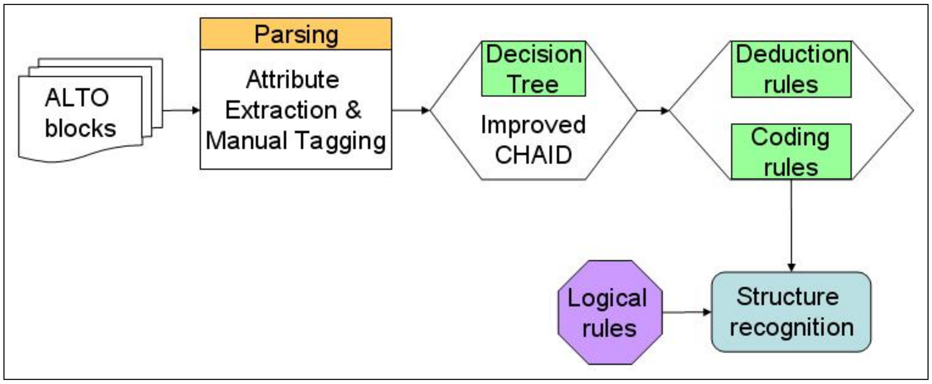


Figure 1. Reverse engineering overview using physical information in ALTO format producing the final logical structure

### 3.1 Feature extraction

Document images are first recognized by an OCR. In this work, we employed the commercial OCR Omnipage 16 from the Nuance company.[13] It is one of the leader softwares to handle OCR and document conversion of home office processes.

On the dataset we used, this OCR reaches almost perfect results in both physical layout analysis and text transcription. There are few errors, and hopefully, only in the character recognition. All the physical attributes of the blocks, that we really rely on, can be always considered as correct.

Note that we concretely use the ALTO (Analyzed Layout and Text Object)[2] XML schema to read our input. Other OCR can be used to replace Omnipage. Most of the time, a XSL is enough to make the conversion from a proprietary format to the standard ALTO.

On top of the individual raw attributes of each block of text, we completed them with some statistics concerning the hole set blocks. This is the case, for example, of the space-size-average which describes the mean empty zone size between each pair of blocks.

The table 2 shows 20 of the 45 possible features for the I-CHAID algorithm. Each line represents a block and each column depicts an attribute.

This table is generated automatically from the ALTO document, then edited manually by the expert to put the logical tag (Title, Author, Address, Abstract, Keywords, SubTitle, Caption, Footer, Other). Our ground-truth is composed of this kind of table.

Table 2. Some block features and their values

| Doc | Hpos | Vpos | Height | Width | Size | Align | Style | WS | Iline | Pag | Pos | CAbs | SWUp | EWP | SWNb | FLUp | Alph | TNb | MTag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 103 | 88 | 19 | 788 | 23 | Center | Bold | 14.0 | 100 | 1 | 1 | false | true | false | false | true | true | false | TITL |
| 1 | 386 | 119 | 24 | 221 | 23 | Center | Bold | 14.0 | 425 | 1 | 2 | false | false | false | false | false | true | false | TITL |
| 1 | 230 | 163 | 14 | 531 | 11 | Center | NULL | 6.0 | 251 | 1 | 3 | false | true | false | false | true | true | false | AUTH |
| 1 | 220 | 198 | 65 | 545 | 10 | Center | NULL | 6.0 | 100 | 1 | 4 | false | false | false | true | true | true | true | ADDR |
| 1 | 312 | 278 | 52 | 364 | 10 | Center | NULL | 6.0 | 100 | 1 | 5 | false | false | false | true | true | true | true | ADDR |
| 1 | 94 | 374 | 278 | 385 | 11 | Block | NULL | 10.0 | 100 | 1 | 6 | true | true | true | false | true | true | false | ABST |
| 1 | 94 | 673 | 44 | 384 | 11 | Block | NULL | 9.0 | 100 | 1 | 7 | false | true | true | false | true | true | false | KEYW |
| 1 | 514 | 374 | 78 | 385 | 12 | Left | NULL | 7.0 | 100 | 1 | 8 | false | false | false | true | true | true | true | STIT |
| 1 | 640 | 487 | 8 | 49 | 7 | Left | NULL | 1.0 | 148 | 1 | 9 | false | true | false | false | true | true | false | OTHR |
| 1 | 724 | 634 | 8 | 86 | 7 | Left | NULL | 5.0 | 143 | 1 | 10 | false | true | false | false | true | true | false | OTHR |
| 1 | 643 | 700 | 6 | 73 | 7 | Left | NULL | 5.0 | 130 | 1 | 11 | false | true | false | false | true | true | false | OTHR |
| 1 | 746 | 529 | 8 | 126 | 7 | Left | NULL | 5.0 | 143 | 1 | 12 | false | true | false | false | true | true | false | OTHR |
| 1 | 95 | 756 | 9 | 178 | 12 | Left | NULL | 7.0 | 100 | 1 | 13 | false | false | false | true | true | true | true | STIT |
| 1 | 94 | 791 | 128 | 387 | 11 | Block | NULL | 9.0 | 100 | 1 | 14 | false | true | false | false | true | true | false | OTHR |

## 3.2 Decision tree construction

The goal is to construct the decision tree which will be as shallow as possible. Indeed, with a deep tree, the decision may be too specific and too close to the learning data. A shallow tree favours the most discriminative attributes and can have high generalisation capabilities. The difficulty lies in choosing among $n$ attributes characterizing the blocks, the one allowing the highest discrimination rate, and having the maximum individuals per node. This process is repeated after each segmentation i.e. each node of the tree. The formal mechanism follows these steps:

- Create the root
- If all the examples belong to the same class or if the number of examples is less than the threshold, then return the class
- Let $a^*$ be the best attribute
- For each value $v$ of $a^*$
  - Add a branch below $a^*$ labelled by $a^* = v$
  - Let $S_v$ the example subset where $a^* = v$
  - Apply recursively the algorithm on $S_v$

The more we go down in the tree, the more complex is the criterion comparison. The number of entries and the number of criteria are also complexity factors. SIPINA allows us to automate the searching process for discriminating criteria performed by $T$.

The figure 2 shows the decision corresponding to the "Abstract" tag. The first node (the root) indicates 4301 blocks where 39 represent abstracts and 4262 other possibilities. These 39 abstracts are then shared into two sets: 10 and 29, thanks to the attribute "Content Abstract" using the segmentation criteria $T$ (equal to 0.6132 in the table 3.a). The other attributes, in the left side, are successively: the relative position in the page ("relativeHpos"), the page number ("page"), and for the right side: the line number ("Nb Line"). These attributes correspond to: "Hpos", "Pag", "NbL" mentioned in the table 2. We can observe that there are uncertainties in the tree concerning the distribution and thus there will be errors during the recognition.

## 3.3 Deduction and coding rules

From this decision tree, SIPINA provides in addition the optimal physical rules to discriminate the sample blocks for each tag. These rules are then translated and coded in a JAVA class model, which aim is to establish the probability that a block will have a specific tag. At each leave of the tree, a correlation index (i.e. $\chi^2$) is performed for each attribute of the father. We consider this correlation index enough discriminant when its value is greater than 0.5. In this case, the condition on the attribute is considered sufficient to discriminate a large number of blocks. It can be seen as an exclusive OR and not as a logical AND which expresses more the child nesting.

The source code stemmed from the previous decision tree (Fig. 2) is given in the figure 4. From the root, we establish the first condition on the block ("isContentAbstractWord") to determine if it is a "Content Abstract"

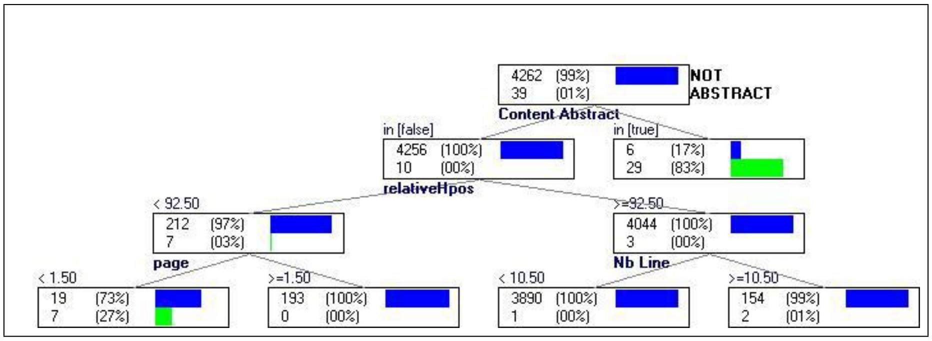
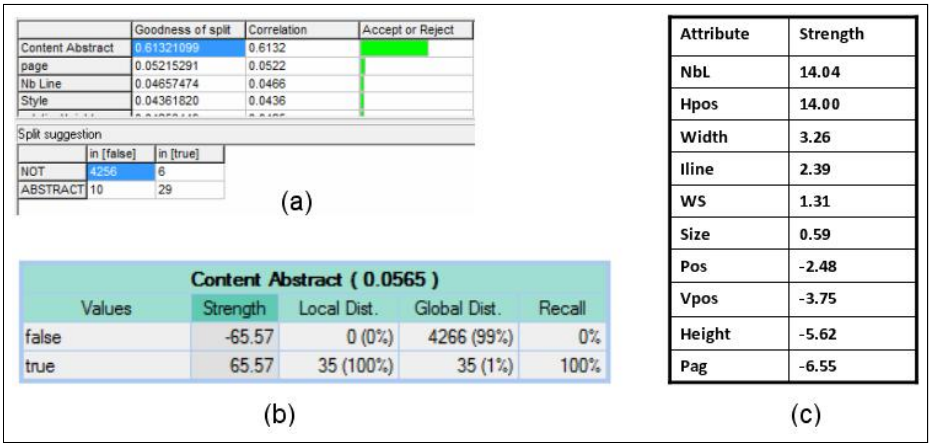


Figure 2. Decision tree of the tag "Abstract"



| | Goodness of split | Correlation | Accept or Reject |
|---|---|---|---|
| Content Abstract | 0.61321099 | 0.6132 | 🟩 |
| page | 0.05215291 | 0.0522 | 🟩 |
| Nb Line | 0.04657474 | 0.0466 | 🟩 |
| Style | 0.04361820 | 0.0436 | 🟩 |

Split suggestion

| | in [false] | in [true] |
|---|---|---|
| NOT | 4256 | 6 |
| ABSTRACT | 10 | 29 |

(a)

Content Abstract ( 0.0565 )

| Values | Strength | Local Dist. | Global Dist. | Recall |
|---|---|---|---|---|
| false | -65.57 | 0 (0%) | 4266 (99%) | 0% |
| true | 65.57 | 35 (100%) | 35 (1%) | 100% |

(b)

| Attribute | Strength |
|---|---|
| NbL | 14.04 |
| Hpos | 14.00 |
| Width | 3.26 |
| Iline | 2.39 |
| WS | 1.31 |
| Size | 0.59 |
| Pos | -2.48 |
| Vpos | -3.75 |
| Height | -5.62 |
| Pag | -6.55 |

(c)

Figure 3. (a) Correlation index, (b) Discrete attribute force, (c) Continuous attribute force

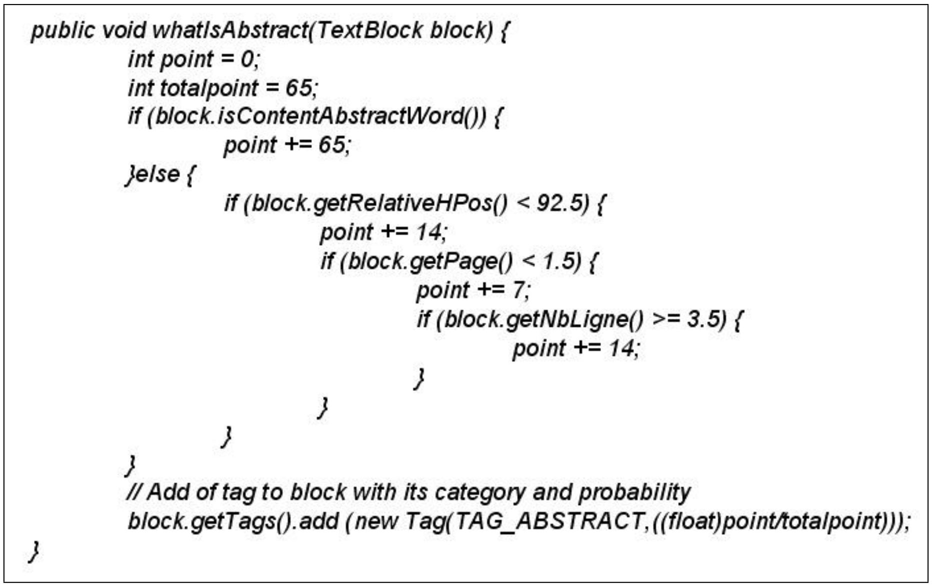
or not. Then, going down in the tree, we iterate with another condition concerning the horizontal position ("getRelativeHeight"), etc. A strength test is performed for each attribute. The corresponding value assigned in the variable point is provided by the strength value of table 3.b and table 3.c. The sum of the point values represents the probability that the block represents an "Abstract".

### 3.4 Logical rules

The logical rules are defined by the operator for a given document class and for each tag. They consolidate and reinforce the tag probability when it is logically well located, and decrease the probability of the other tags which do not correspond to the logical model. The table 3 indicates the correction weights (translated in probabilities). These gains or corrections are set empirically.

Table 3. Reinforcement and weakening of the tag probabilities

| Tag | Father | Gain | Son | Gain |
|---|---|---|---|---|
| Abstract | Abstract/Address | +2 | Abstract/Keyword | +2 |
| | Otherwise | -2 | Otherwise | -2 |
| Address | Address/Author | +2 | Address/Abstract | +2 |
| | Otherwise | -1 | Author | +1 |
| Author | Author/Title | +2 | Author/Address | +1 |
| | Otherwise | -2 | Otherwise | -2 |
| Caption | Others/Sect_Title | +1 | Others/Sect_Title | +1 |
| Keyword | Abstract | +2 | Keyword/Sect_Title | +1 |
| Footer | Others/Sect_Title | +2 | Null | +1 |
| Sect_Title | Others/keyword | +1 | Others/Sect_Title | +2 |
| | Sect_Title | +2 | Reference | +1 |
| | Otherwise | -3 | Otherwise | -3 |
| Title | Null | +2 | Title/Author | +2 |
| | Title | +2 | | |
| | Otherwise | -1 | | |
| Others | Others/Sect_Title | +2 | Others/Sect_Title | +2 |
| | Otherwise | -4 | Otherwise | -4 |



```
public void whatIsAbstract(TextBlock block) {
        int point = 0;
        int totalpoint = 65;
        if (block.isContentAbstractWord()) {
                point += 65;
        }else {
                if (block.getRelativeHPos() < 92.5) {
                        point += 14;
                        if (block.getPage() < 1.5) {
                                point += 7;
                                if (block.getNbLigne() >= 3.5) {
                                        point += 14;
                                }
                        }
                }
        }
        // Add of tag to block with its category and probability
        block.getTags().add (new Tag(TAG_ABSTRACT,((float)point/totalpoint)));
}
```

Figure 4. Source Code generated from the ”Abstract” decision tree

### 3.5 Structure recognition

Two methods have been experimented. The first one uses a decision tree per tag. For each block, the physical and logical rules are applied. Each rule stemmed from each decision tree provides a specific tag which leads to a multi-tagging for each block. The tag with the highest strength is selected as the winner. The tag strength is the weighted sum of the physical and logical probabilities. These probabilities are weighted considering the empirical ratios 2/3 for the physical and 1/3 for the logical.

The second method uses a single decision tree which is global to all the tags. As previously, for each block, corresponding physical and logical rules are applied. Each block is labelled only once.

## 4. EXPERIMENTS AND DISCUSSIONS

We have experimented this approach on around 1000 documents containing about 16,000 blocks. They are stemmed from ICPR04[7] and ICPR08[8] proceedings. The documents are in PDF and can be directly read by Omnipage OCR 16. The two approaches are compared using the measures: recall, precision, insertion rate and error rate. Globally, the results are satisfactory. In the first approach (i.e. with one tree by tag), the average recall is around 95.7%, the average precision is around 93.5% and the average error rate is 5.9%. In the second approach (i.e. with one global tree), the average recall is around 93.0%, the average precision is around 94.0% and the average error rate is 6.4%. Figure 5 details the percentages for each tag. The chart on the left side shows the precision and recall measurements of both methods while the chart on the right side shows error rates for both methods. The multi-tagging method is more flexible to add new tags type and obtain better results. However the results of both methods are close.

## 5. CONCLUSION AND PERSPECTIVES

We have presented in this paper a new method for logical document structure labelling. It is based on decision tree which learn the relationships between the physical features given by commercial OCR on each block of text and the logical labels given by an expert during the training step. The decision tree is consolidated by logical rules, also given by the expert, which are simply deduced from the hierarchical organisation of the blocks. Although all the different steps of the system are not yet fully automatic, we conducted experimentation on 1000 real documents picked from the ICPR 2004 and 2008 proceedings. The first results showed that the error rates are in average less than 6% for the 9 possible labels.

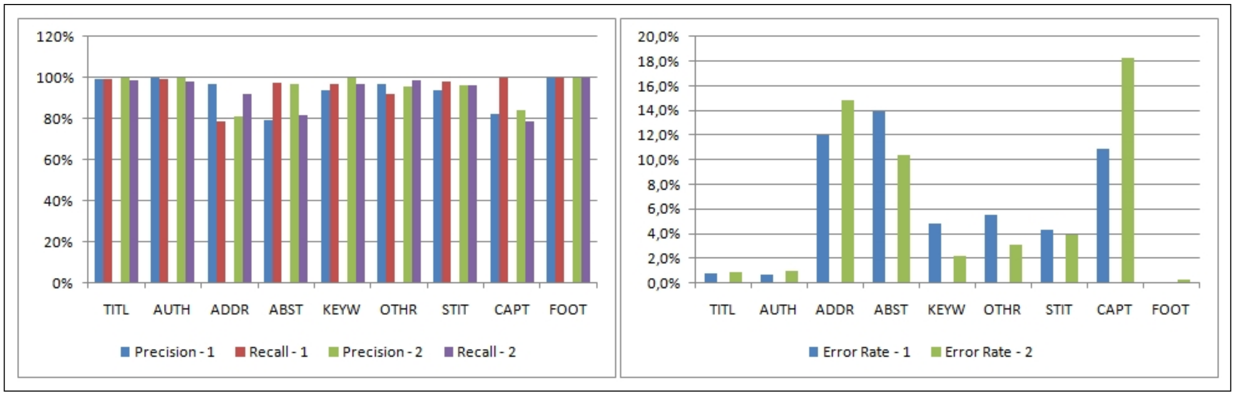


Figure 5. Result comparison between the two approaches.

## REFERENCES

[1] Akindele O. T., Belaïd A. (1995) Construction of Generic Models of Document Structures using Inference of Tree Grammars International Conference on Document Analysis and Recognition (ICDAR), pp 206-209 ICDAR'95, Montral, Qubec, August 14-16, 1995.

[2] ALTO. http://www.loc.gov/standards/alto/

[3] Bapst F. (1998) Reconnaissance de documents assistée : architecture logicielle et intégration de savoir-faire PhD thesis, IIUF-University of Fribourg, Fribourg, Switzerland 1998.

[4] Belaïd A., Rangoni Y., Falk I. (2007) XML Data Representation in Document Image Ananysis. International Conference on Document Analysis and Recognition. (ICDAR), Vol. I, pp 78–82, Curitiba, Brasil, Sept. 2007.

[5] Chauchat J. H., Rakotomalala R., Carloz M., Pelletier C. (2001) Targeting customer groups using gain and cost matrix: a marketing application, Proc. of Data Mining for Marketing Applications Workshop, PKDD'2001, pp. 1-13, 2001.

[6] Dori D., Doermann D., Haralick R., Ihsin P., Buchman M., Ross D. (1995) The Representation of Document Structure, a Generic-Object Process Analysis Handbook on Optical Character Recognition and Document Image Analysis, Eds P. S. P. Wang and H. Bunke, World Scientific Publishing Company, 1995.

[7] ICPR'04. http://www.ee.surrey.ac.uk/icpr2004/

[8] ICPR'08. http://www.icpr2008.org/

[9] Kass G. (1980) An exploratory technique for investigating large quantities of categorical data, Applied Statistics, 29(2), 119-127, 1980.

[10] Morgan J. N., Sonquist J. A. (1963) Problems in the analysis of survey data, and a proposal JASA, Vol. 58, n 302. Zbl 0114.10103. 1963.

[11] Morgan J. N., Messenger R. (1973) THAID-a sequential analysis program for the analysis of nominal scale dependent variables, Survey Research Center, U of Michigan, 1973.

[12] Nagy G. (2000) Twenty Years of Document Image Analysis in PAMI. IEEE Trans. Pattern Anal. Mach. Intell. 22(1): 38-62 (2000)

[13] Nuance. http://www.nuance.com/imaging/products/omnipage.asp.

[14] Quinlan R. (1979) Discovering rules by induction from large collections of examples, D. Michie ed., Expert Systems in the Microelectronic age, pp. 168-201, 1979.

[15] Rakotomalala R. (1997) Induction Graphs, PhD Thesis, Universit Claude Bernard Lyon 1, 1997.

[16] Terrenoire M. (1970) Un modèle mathématique de processus d'interrogation: les pseudoquestionnaires, PhD Thesis, Université de Grenoble, 1970.