A Structural Classifier to Automatically Identify Form Classes

P. Héroux[1], S. Diana[1,2], E. Trupin[1], Y.Lecourtier[1]

[1]Laboratoire PSI, Université de Rouen, Place E. Blondel,
76821 Mont Saint Aignan Cedex, France
{Pierre.Heroux, Sebastien.Diana, Eric.Trupin, Yves.Lecourtier}@univ-rouen.fr

[2]DPCi S.A., 15 rue J-B Colbert BP 6042,
14062 Caen Cedex

Abstract. This article deals with the description of a new classifier for an automatic form class identification system. This new structural classifier is based on a tree comparisons. The high level information used by this classifier is presented in the article. A module first extracts the form content. The form content organisation is described in a hierarchical way modelled by a tree. This tree corresponds to the input features of the structural classifier. Experimental results are presented and several strategies of combined uses of this structural classifier with other classical classifiers are suggested in order to enhance the results.

1. Introduction

A form processing system extracts and automatically interprets the content of the forms. The principle of a such system is based on the knowledge of the location of some areas on the form and their meaning. Active areas correspond to hand-filled parts of the form. The list of areas, their location, their meaning and the consistency links describe the reading model. Therefore, form processing systems contain one reading model per form class, and, in order to process a form from a given class, it is necessary to specify the corresponding reading model.

Our approach is based on a preliminary automatic determination of the form class to select the associated model, and then process the form. Thus, the form processing system is upstream completed by a form class identification module (Fig. 1).

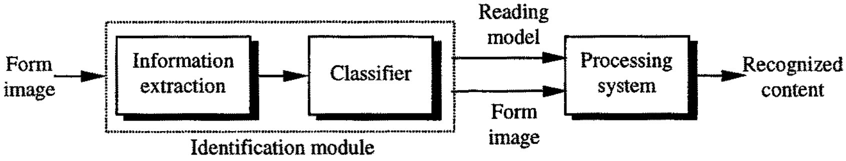


Fig. 1. Evolution of the form processing system

Our identification module is composed of two parts. First, the module extracts a structural information (tree) representing the form content organisation. Then, this information is used by a new structural classifier.

This paper describes the new structural classifier. Section 2 presents the structural information extraction module. Section 3 deals with the structural classification method based on tree comparison. Experimental results are detailed in section 4. Finally, strategy prospects of uses of this structural classifier combined with other classical classifiers are presented in section 5 as a conclusion.

2. Hierarchical Structure Extraction

The information extracted from the binary form image is a high level representation of the form content organisation. The result of the extraction process describes the form content organisation with a tree representing the hierarchical dependency of the different elements of the form.

The module is organised in five main processes (Fig. 2). It extract a set of features from the binary image (list of blocks with different information) which are then organised. Finally, the form content is represented by a tree.

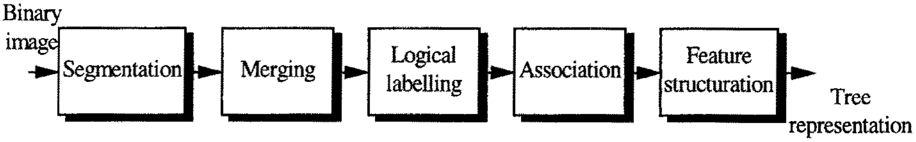


Fig. 2. Hierarchical structure extraction module

A segmentation process which extracts homogeneous blocks containing 8-connected objects is applied [1] [2] on the image. Different 8-connected object categories permit to give a layout label to each block : text, graphic, straight line and table. The blocks which are labelled as text blocks are then cut into lines in order to determine the number of the lines which compose them. The relative location of the lines permit to perform a logical labelling (paragraph begin, paragraph, paragraph end,...) of these text blocks. The set of the extracted elements are organised to obtain a tree in which each node is composed by several features (location, dimension, layout label, logical label, text line height, number of sons).

Fig. 3 shows the list of extracted blocks on a sample of form and Fig. 4 shows the corresponding tree created with extracted blocks. The segmentation process tends over-cut the forms, therefore, neighbouring blocks with same categories are merged into meta-blocks. For more details on this structure extraction module, refer to [1] and [2].

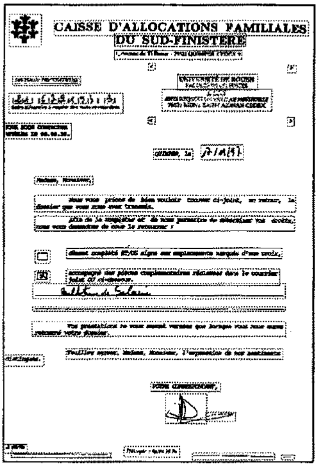


Fig. 3. List of extracted blocks

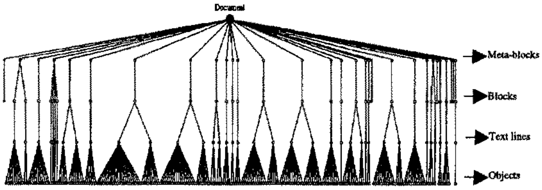


Fig. 4. Tree representation

3. Structural Classification Method

This method permits the form class identification. It is based on comparisons between the tree representing this form and the trees characterising the form classes (model trees). The hierarchical structure of the form, extracted by the information extraction module presented in section 2, is compared with the model tree of each form class. Tree properties [5] are used to perform this comparison considering that i) the tree is a particular graph and ii) the tree is a set of nodes where R is the root and the other nodes, root of sub-trees linked to R with edges. This recursive properties of trees are used in dynamic programming. Different measures can be defined to characterise the similarity between two trees. Some of them are based on isomorphism and label similarities of nodes and edges [6]. Other techniques concern the graph-matching [7]. However, these methods do not use the recursive properties.

The Selkow's algorithm [5] specifically concerns tree comparison problems. This is a generalisation of the method defined by Wagner and Fisher concerning distance between strings called edition distance. The trees are labelled. Three operations are used to transform an input tree into an output tree : sub-tree substitution, sub-tree insertion and sub-tree deletion. A cost is attributed to each elementary operation. To transform a tree into an other one, many sequences of elementary operations are can be followed. The Selkow's distance between two trees is then established by following the sequence of elementary operations leading to the minimal cost. The costs associated to the different operations have to be fixed so that they express the subjective human perception. The main problem of this algorithm concerns its computation time because of the dynamic programming and the difficulty to determine adapted costs.

Thomasson and Gonzalez [8] present an other measure of similarities between trees, but this measure is more restrictive. Because of its definition, variations of node position are not allowed. To solve our problem, an improvement of this method is proposed. It consists in an iterative computation :

- The roots of the compared trees are examined. If they are equal, we look for equal nodes among their sons.
- This operation is repeated on pairs of sub-trees whôse roots are equal.

Besides, two nodes are considered as equal if the difference between each attribute of the nodes does not exceed a threshold. The attributes are location, dimension, layout and logical labels. The thresholds have been determined after a statistical study of a large set containing pairs of equal nodes. Otherwise, the fact that two nodes do not have to be at the same position in the trees allows to modify the tree-distance defined by Thomasson and Gonzalez. Finally, this algorithm returns a tree which is common to the compared trees. This algorithm is used in the form class identification (section 3.1). Section 3.2 presents model tree construction. An organisation of the model tree base which improve identification is developed in section 3.3.

3.1 Form Class Identification

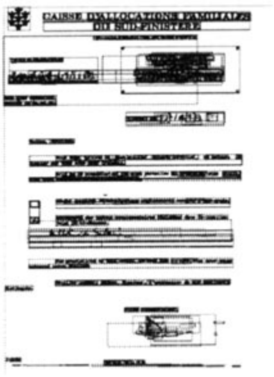
To identify the class of a studied form, comparisons between the tree representing the form and the model trees representing the classes are performed. Each comparison returns the common tree between the studied tree and a model tree. Three features are then extracted after each comparison : the number of nodes in the common tree and the two overlapping rates between the common tree and the compared trees. These proportions are used to limit the density variations influence between classes and the low stability problem in each class. The three features characterise the quality of the comparison. The examination of these features allows to determine which class is the nearest of studied form. When the class is identified thanks to the tree comparison, the result is confirmed by submitting the features characterising the quality of the comparison to a threshold. If the features do not satisfy the threshold, the result is rejected considering that the form does not belong to any of the known classes. The definition of the threshold is presented in the next section.

3.2 Model Tree Construction

Each form class is represented by a model tree. This model tree includes the frequently encountered features in a given form class. The model tree construction also takes into account differences with other form classes. A model tree must be built to present a great number similarities when compared with trees representing form from this class. Moreover, it has to present very few correspondences with trees of other classes. Thus, the main difficulty of the construction of such trees is the necessity to consider the form variability inside a class.

Studied documents are filled up forms and hand-written data are added in specific data capture areas. The variability of these data involves a low stability for the corresponding nodes contrary to some passive areas. Moreover, the block dimension have an influence on the stability of the corresponding nodes. This induces the creation of model trees during a training phase to characterise these specific nodes. First, a list composed of the most stable nodes is extracted from the trees of the training set. Each node of each tree is compared with nodes of the same level in the other trees. For each node, its appearance frequency is determined in order to reject those which are not significant enough. In a second step, the model trees are built with the stable nodes list by linking nodes of two consecutive levels which present compatible labels (for example, a graphic block can no be included in a text block).

After this tree model construction, the last step determines an adaptive threshold for each class. For each modelled class, features characterising the quality of comparisons between the model tree and the trees from the training set are considered. The distribution of these features is Gaussian. The mean *m* and the standard deviation *sd* can be calculated. The identification process will return the nearest class for a studied form but only if the feature characterising the quality of the comparison is above a threshold equal to m-2*sd*. These adaptive thresholds present the interest to be automatically and independently determined for each class after the construction phase of the model trees. They allow to reject unknown forms.


CAISSE D'ALLOCATIONS FAMILIALES
DU SUD-FINISTERE

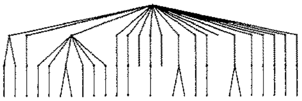


Fig. 6. Tree model obtained

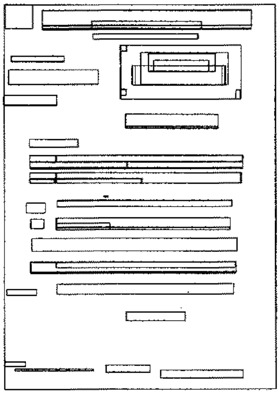


Fig. 5. List of stable blocks

Fig. 7. Model representation

An example of model tree corresponding to the form class of the Fig. 3 is presented on Fig. 6 as the associated list of stable blocks (Fig. 5) and the representation on a filled up form (Fig. 7).

3.3 The Hierarchy of Models

The tree comparison method is improved by an organisation of the model trees base which reduces the computation time. This organisation correspond to the construction of a hierarchy of model trees.

Among the stable nodes, there are nodes which are common to several classes. These nodes represent, for example, a logo or a headband. These correspondences between classes are extracted to define *meta-models*. These meta-models define the common representation of several classes. The hierarchy is built by recursively grouping model trees and meta-models into meta-models. The hierarchy representation corresponds to a binary tree where the non-terminal nodes are meta-models and the terminal nodes are the models (Figure 8).

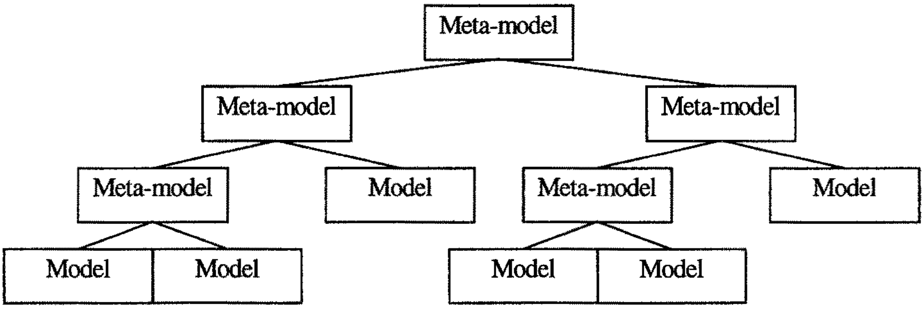


Figure 8 : Hierarchy representation

The identification is based on tree comparison. It consists on finding the path of the hierarchy that leads to a terminal node (class model). For each node of the hierarchy, the tree of the studied form is compared with the two models or meta-models which are the sons of the current node in order to go down to the best correspondence. To make the method more reliable, the k best correspondences search algorithm is used. This permits to explore simultaneously the k most probable paths.

This improvement involves a lower number of comparisons than for the whole model base exploration when the number *n* of classes increases ($n > 2.k.\log_2(n)$). Moreover, a comparison with a meta-model is faster than with a model tree because a meta-model is composed only with the common nodes of two models or meta-models.

4. Experimental Results

The results have been established with a 1420 form set. This set was cut in a first set (TB1) containing 1300 forms divided into 26 known classes of 50 forms and in a second set (TB2) containing 120 forms divided into 12 unknown classes of 10 forms. Otherwise, the training phase has been implemented with the tree model construction with 10, 20, 30 or 40 filled up forms per class.

The results (Table) show a good capacity to reject the forms from unknown classes and no error during the identification of the known class forms with an interesting recognition rate.

The number of forms used in model tree construction seems to be an important information because the recognition rate increases and the reject rate decreases with the number of forms in the training set. It permits to highlight the stability of the information contained in the forms and the variability added by hand-written text.

This classifier gives good results, but it must be validated with a bigger number of forms and with a base which contains more classes to study the behaviour of the error rate when the number of classes increases.

Computation time has been tested on the identification with the hierarchy and with the simple tree matching. The form processing system has been implemented on a SUN Sparc 20 station. The form identification on 15 model trees takes 540 ms for simple tree matching and 500 ms with the model hierarchy. The difference is not significant, but it should increase when the model base will be larger (120 models).

Number of learned	Known classes (TB1)			Unknown classes (TB2)	
trees per class	Recognition	Reject	Error	Reject	Error
10	87.31 %	11.54 %	1.15 %	100 %	0.00 %
20	94.62%	5.38 %	0.00%	100 %	0.00 %
30	97.31 %	2.69 %	0.00 %	100 %	0.00 %
40	99.23 %	0.77 %	0.00 %	100 %	0.00 %

Table : Results of the structural classifier

5. Conclusion

We presents in this paper the description of a form class identification module. This module is based first, on the extraction on the binary image of a hierarchical structure (modelled by a tree) representing the form content organisation, and secondly, on a classifier module which recognises the class of the form.

The classifier presents a good recognition rate (tested on 1420 forms divided up into 38 classes). The recognition rate reaches 99.23 % with 0.00% for the error rate. These results illustrate the very good robustness of the used features. However, it is necessary to evaluate the performance of the classifier on a larger test set (more forms and more classes). This module will be integrated in an application of automatic form processing with 120 different form classes.

Our prospects are based on the set up of a classification strategy in order to use this structural classifier with classical classifiers (k-Nearest Neighbours [3] or Multi Layer Perceptron [4]). The aim corresponds to use classical classifiers which seem to be faster, but to use our structural classifier which has better reject rate when the classes are unknown (see Table : Results of structural classifier).

Two strategies are envisaged. The first strategy corresponds to use the classical classifiers as pre-classifiers to reduce the number of candidate form classes. Thus, the structural classifier will be applied on the reduce list to identify the correct class thanks to the amount of information contained in the hierarchical tree representation. The second strategy corresponds to establish a co-operation between the different classifiers (classical and structural). Each classifier suggests its reduced list of form classes and the correct class will be selected by vote. Obviously, the employed strategy will depend on the experimental results of the different classifiers on a base of 120 form classes.

Acknowledgements

This work was developed in the ADI (*Intelligent Archivage and Documentation*) project context in association with the DPCi Company. The DPCi Company is specialised in electronic printing, scanning and document processing

References

1. Diana S., Trupin E., Lecourtier Y., Labiche J.: An Assistant to the Modelisation of Forms. MultiMedia Signal Processing Princeton NJ USA (1997) 163-168
2. Diana S., Trupin E., Lecourtier Y., Labiche J.: From Acquisition to Modelisation of a Form Base to Retrieve Information. International Conference on Document Analysis and Recognition Ulm Germany (1997) 762-765
3. Cover T.M., Hart P.E.: Nearest Neighbour Pattern Recognition. IEEE Trans. on Information Theory Vol.13 **1** (1967) 21-27
4. Rumelhart D.E, McClellard J.L.: Parallel Distributed Processing : Explorations in the Microstructure of Cognition. Foundations, The MIT Press Vol.1 (1986)
5. Miclet L.: Méthodes structurelles pour la reconnaissance des formes. (ed.): Eyrolles (1984)
6. Ishitani Y.: Model Matching Based on Association Graph for Form Image Understanding. . International Conference on Document Analysis and Recongnition Montreal Canada (1995) 287-292
7. Budin A.: On The Problem of Attributed Relational Graph Matching. Automatika **33** (1992) 151-157
8. Thomasson M.G.:, Gonzalez R.C: Syntactic Recognition of Imperfectly Specific Patterns. IEEE Trans. on Computers Vol.24 **1** (1975) 93-96