A RULE-BASED SYSTEM FOR DOCUMENT UNDERSTANDING

Debashish Niyogi and Sargur N. Srihari

Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260, USA

ABSTRACT

A rule-based system to make inferences about document images is introduced. Given a digitized document image, the system controls the analysis of the document, and identifies all the different printed regions in the document image. Logical "blocks" of information on the document image are interpreted and classified by this system which then produces as output an editable description of the entire document. The system uses a goal-directed top down approach, and utilizes a three-level rule hierarchy to implement its control strategy.

1. INTRODUCTION

Document understanding is a task that is analogous to speech understanding and image understanding. A document, more specifically a printed document, has printed text, line drawings, half-tone pictures, graphs, icons, etc. As a domain for serious research, document understanding has been gaining in importance over the years.

Early interest in this field was primarily because of the need to store and transmit large volumes of information that are contained in documents. A need was felt to be able to code the information in the documents, and then to store/transmit this code such that the document image can then be reconstructed at another site. Several document encoding techniques have been developed for this purpose. Efforts have also been made to develop new techniques for analyzing individual components of a document with a view to deciding whether a given component is composed of text or graphics. Different segmentation techniques have been proposed, and used, to varying degrees of success.

Relevant work in this area include the use of a non-linear run-length smoothing algorithm for the segmentation and classification of digitized printed documents into regions of text and images [Wong, Casey and Wahl, 1982]. A survey on document image analysis [Srihari, 1986] gives a comprehensive overview of known techniques in all aspects of document image analysis. A discussion of techniques used in analyzing pieces of letter-mail to locate the destination address can be found in [Srihari et al, 1985].

A more recent interest in document understanding is that of trying to design systems which embody knowledge about the basic structure of different kinds of documents and use this knowledge to analyze and identify the different components of a document. Such a system would tie in together various aspects of document image analysis, like edge segmentation, filtering, etc. along with a high-level control structure that interprets the document image with the help of these image processing operations. A knowledge-based system that can direct the classification of the different entities on a document image and decide when an unambiguous classification of all the relevant entities has been achieved, is one of the major goals in the field of document understanding.

Knowledge-based systems have been used in the past in various domains [Barr and Feigenbaum, 1982]. One of the first knowledge-based systems was MYCIN, which illustrated how knowledge gleaned from experts could be represented in the form of production rules that could be used in medical diagnosis [Buchanan and Shortliffe, 1984]. The use of knowledge-based systems for image analysis include an expert system for low level image segmentation of visual scenes [Nazif and Levine, 1984] and a rule-based system for aerial imagery [McKeown, Harvey and McDermott, 1985]. Rule-based strategies for image interpretation have been proposed in [Weymouth, Griffin, Hanson and Riseman, 1983], and a knowledge-based computer vision system has been described in [Levine, 1978]. The application of knowledge-based techniques to document image understanding have been discussed in [Kubota, Iwaki and Arakawa, 1984], which describes the application of a production system concept to an experimental document understanding system, and more recently in [Nagy, Seth and Stoddard, 1985] which proposes the use of X-Y trees for the representation of information about a document image.

We propose here a knowledge-based system that is organized as a production system with different levels of production rules that perform an analysis of a document image, and interpret and classify the various regions of printed matter on the document. The input to this system is a digitized document image, and the output is an editable description of the document.

This paper first describes (in Section 2) the overall architecture of the system, and then gives details about the various components of the system. The innards of the knowledge base, the control structure, the inputs to the system and the outputs of the system are explained. Section 3 describes techniques by which the system deals with uncertainty. In Section 4, some actual rules (in Prolog) used in the system are shown and explained. Section 5 describes actual results obtained so far using the rule-based system. A discussion on the applicability of rule-based systems to the document understanding problem follows in Section 6.

2. ARCHITECTURE OF THE SYSTEM

The knowledge-based system that we are developing is composed of two basic parts : the Knowledge Base and the Control Structure. The input to the system consists of the document image data. The Control Structure (Inference Engine) uses the knowledge contained in the Knowledge Base along with its control strategy to make inferences about the document from the given image data. The output of the system is a descriptive classification of the various identifiable printed regions, or *blocks*

This work was supported in part by the United States Postal Service Contract 104230-85-M3349 and by the Xerox Webster Research Center.

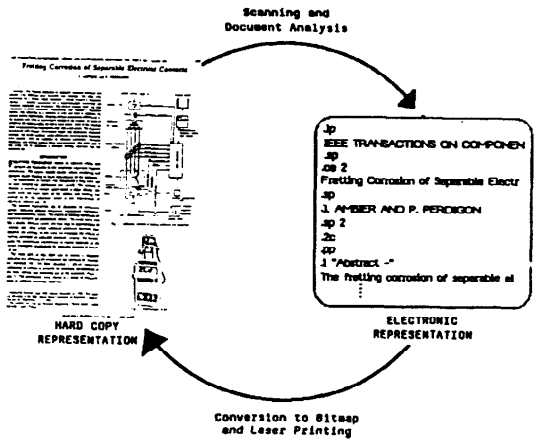

Fig. 1 System input (a document image) and output (an editable description)

in the document image. Figure 1 shows a sample document image that is input to the system and the editable description that is output by the system. (As Figure 1 indicates, it is possible to "reconstruct" the document from its editable description; however, the current system provides no facilities for this, but rather concentrates on the problem of analyzing the original image to obtain the description.)

The control flow in the overall document understanding system is as follows : the document is first digitized and the resulting digital image is segmented to obtain data about the various printed regions in the document. This data includes the intrinsic properties (e.g., shape, size, aspect ratio, etc.) of each of the identified regions, as well as the spatial relationships between the various identified regions in the document image. The control structure then uses the knowledge base to examine this data, and attempts to arrive at a consistent classification for each of the identified regions, or blocks.

The system consists of three levels of rules : Knowledge Rules, Control Rules and Strategy Rules. The knowledge rules contain knowledge about the intrinsic properties of the various regions of a document image, and also the spatial relationships between these regions. The control rules decide what knowledge rules are to be executed and in what order, and thus act as focus-of-attention mechanisms to guide the search towards a more efficient resolution. The strategy rules supervise the entire search and classification process, and determine what control rules are to be executed at any given time and in what order. Strategy rules also determine whether a consistent interpretation of the image has been obtained.

If the data from the initial segmentation of the image is not sufficient for an unambiguous interpretation of the document image, then the system decides to obtain more data from the given image. Thus, any further image processing operations that are required are progressively invoked under the supervision of the inference engine. These operations could include further segmentation of the image, color filtering, text reading, etc.

A goal-driven (top-down) approach is used by this system, which uses a hypothesize-and-test strategy for arriving at its conclusions. Thus, the system makes hypotheses about different intermediate conclusions and chains backwards through the rules in order to test the hypotheses. In trying to satisfy a hypothesis, some other hypotheses may be generated which must first be tested before the original hypothesis can be considered to be justified. Thus, an entire set of backward-chaining processes are set up, and the system only reaches a satisfactory conclusion when all these processes have run to completion.

2.1. INPUTS TO THE SYSTEM

As mentioned above, the initial input to the knowledge-based system consists of data obtained from low-level image segmentation and filtering performed on the original document image. This data is composed of two parts : Descriptions of the characteristics of each region of printed matter on the image, and a Relational Data Structure which represents the spatial relationships between each of these regions. The regions identified by the initial segmentation and filtering process may not, however, always represent the logical *blocks* that the knowledge-based system would attempt to classify. Thus, an intermediate step is necessary to combine/ split up these regions as necessary so as to arrive at the logical blocks required by the rule-based system. This intermediate step is known as *Region-Merging*. This step essentially combines individual letters (connected components identified by the initial segmentation) into words, lines and finally paragraphs (logical "blocks"). The outputs of the region-merging process are the descriptions and relational data structures for the logical printed blocks in the document image.

2.2. THE KNOWLEDGE BASE

The knowledge base for this system consists of a set of rules that embody knowledge about the general characteristics about document images. These rules are expressed in terms of predicates in first order predicate logic. The rules in the knowledge base are called *Knowledge Rules*. These rules define the general characteristics expected of the usual components of a document image, and the usual relationships between such components in the image. For example, in a document like the cover page of a journal article (shown in Figure 2 (a)), the various blocks of printed matter correspond to the journal banner, the title of the article, the names of the authors, the abstract, section headings, various paragraphs, perhaps one or more line drawings or figures or tables, footnotes, etc. (These different blocks are shown in Figure 2 (b)). Also, the usual relationships, e.g., the title being above the author names, the abstract being above the first paragraph of text, the footnotes being at the bottom of the page, etc. are generally true of such documents. Intrinsic properties, like the block-to-white pixel ratio for half-tone figures in the image being larger than the corresponding ratio for text, are also true in general for such blocks. From such known facts about these kinds of document images, rules are constructed that can be used by the inference engine to make inferences about the various identified "blocks" on the given document image.

2.3. THE CONTROL STRUCTURE

The control structure for the rule-based system consists of an inference engine which uses the knowledge base to make unambiguous inferences about the classification of various blocks in a given document image. The inference engine is also rule-based, and contains two levels of rules : *Control Rules* and *Strategy Rules*. These rules regulate the analysis of the document image, and decide when a consistent interpretation of the image has been obtained. The inference engine uses a top-down approach in arriving at its solution, since the solution space is not very large, and a lot of knowledge exists (in the knowledge base) about the domain. A backward-chaining process is used by the control structure.

The rules comprising the inference engine are also coded in terms of predicates in first-order predicate logic. The control structure determines the order in which these rules are executed in order to test various conditions effectively. Control rules can

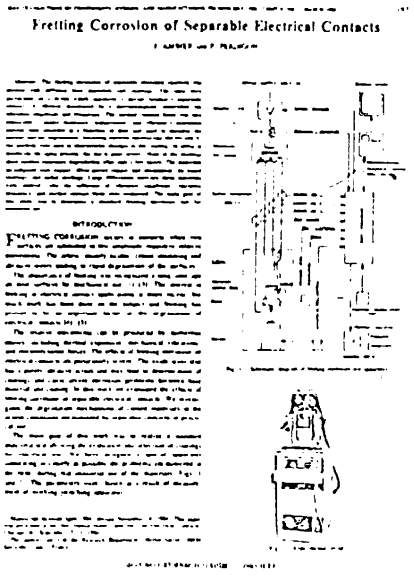

Fig. 2 (a) A sample document

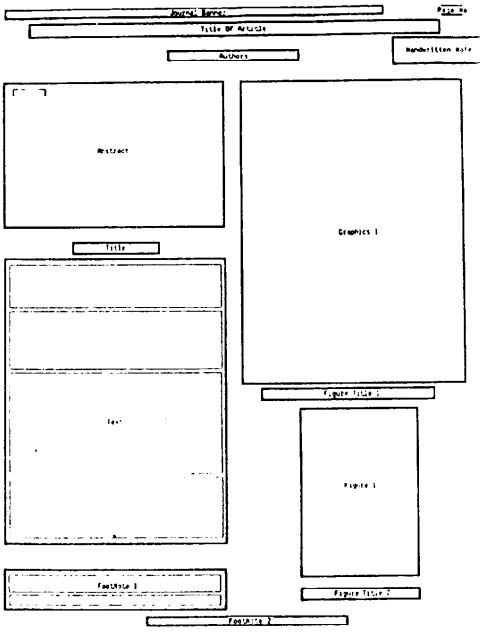

Fig. 2 (b) Logical blocks identified in the sample document

be "focus-of-attention" rules or "meta-rules". For example, at any given stage of the analysis, control rules can decide that all the relevant knowledge rules for footnotes be executed so as to test whether the given block, say b1, is a footnote. Strategy rules can guide the search in a more general way, i.e., they can determine what strategy is to be followed at any given time for analyzing the image. This means that the strategy rules determine what the order of execution of the control rules will be.

It is important to note here that the control structure of the system is actually more than just a set of production rules. Prolog, which is used to implement the rule-based system, has been used here as a sophisticated programming language, and various intricate features of this language have been put to use for monitoring and controlling the different levels of production rules.

3. REPRESENTATION OF UNCERTAINTY

The system has to deal with many situations where a combination of rules, rather than a single rule, lends credence to a particular hypothesis. Thus, the success of each of these rules adds evidence towards that hypothesis. If the total evidence obtained from the successful rules is sufficiently high, then the hypothesis is assumed to be true, and the next stage in the analysis process can then be tackled with the assumption that the given hypothesis has been confirmed. To deal with such a scenario, each Knowledge Rule in the system is given a certain "confidence value" between 0 and 1. When the knowledge rules for testing the characteristics of a certain type of block are executed, the confidence values for all the rules that succeed are added up. The sum thus obtained indicates the "certainty factor" for the conclusion obtained from the control rule which invoked these knowledge rules. This certainty factor is used for the purpose of ordering the conclusions at any given stage so that the more likely conclusions can be examined in further detail *before* the less likely ones. This has the effect of making the search process more efficient, thus reducing execution time in the system.

One possible inconvenience in this representation is that when a set of new rules are added to the system, the relative importance of the existing rules may change, and thus the "confidence value" for the existing rules may have to be altered. To counter this problem, an even more robust scheme for the representation of uncertainty is currently under consideration. This scheme is based to a large extent on the representation used in the DGMES scheme proposed in [Wesley and Hanson, 1982]. In this representation, the evidence for a hypothesis is maintained in two parts : the Support (evidence in favor of the hypothesis) and the Plausibility (evidence against the hypothesis). The evidence is thus maintained as a pair [SUPT,PL] where SUPT = Support and PL = Plausibility. The confidence value associated with any rule can therefore be a positive or a negative number. Initially the SUPT value for a control rule is 0, and the PL value is 1. When the knowledge rules are executed, the positive evidence is added to the SUPT value, and the negative evidence is subtracted from the PL value. The final analysis of the hypothesis is thus done based on both the evidence in favor of the hypothesis as well as the evidence against it. This scheme has the advantage of being able to distinguish between *negative* evidence and *no* evidence. Implementation of this uncertainty representation scheme is planned in the next stage of the system development process.

4. THE RULES

The rule-based system has been implemented in Prolog, which has built-in machanisms for backward chaining through its rules that are expressed as predicates in first-order logic [Clocksin and Mellish, 1981]. As mentioned above, the production rules used in this system are of three kinds : Knowledge Rules, Control Rules, and Strategy Rules. Examples of each of these kinds of rules are given in Figures 3, 4 and 5. Brief explanations of these rules are given below. Detailed explanations of the Prolog code have been avoided for the sake of brevity, but can be obtained by referring to (Clocksin and Mellish, 1981) or any other book on Prolog.

```
isblock('dest-address',B) :-
                        color([K1,K2,K3]),
                        bl_and_wh(K1,K2,K3),
                        B is 0.25.

bl_and_wh(X,[[H1|T1],[H2|T2],[H3|T3]]) :-
                                    extract_b_w(T3).
extract_b_w([H4|T4]) :- H4 == 0.
```

Fig. 3.1 Unary Knowledge Rules

```
block('dest-address',A,X1) :-
                        block('postage-stamp',B,X2),
                        left_of(A,B),
                        below(A,B),
                        X1 is 0.3 * X2.
```

Fig. 3.2 Binary Knowledge Rule

```
findblock(X,Y) :-   findall(Z,isblock(X,Z),L),
                    addup(L,Y).

addup([],Y) :-      Y = 0.
addup([H|T],Y) :-   addup(T,Z),
                    Y is H + Z.

findall(X,G,_) :-   asserta(found(mark)),
                    call(G),
                    asserta(found(X)),
                    fail.

findall(_,_,L) :-   collect_found([],M), !, L = M.

collect_found(S,L) :- getnext(X), !, collect_found([X|S],L).
collect_found(L,L).

getnext(X) :-       retract(found(X)), !, X == mark.
```

Fig. 4 Control Rules

```
begin :- pstr("What type of block would you like to identify ? "),
        read(X), X == end_of_file,
        decide(X),
        begin.

decide(X) :- datablocks(Z),
        try(X,Z).

try(X,[]) :-   fail.
try(X,[H|T]) :- tryone(X,H).
try(X,[H|T]) :- try(X,T).

tryone(X,Y) :- reconsult(Y),
               identify(X,Y).

identify(X,Y) :- findblock(X,Z), nl,
                 pstr("The "), write(X),
                 pstr(" is : "), write(Y),
                 pstr(" with certainty = "), write(Z),nl,nl.
```

Fig. 5 Strategy Rule

Figure 3.1 shows a set of unary knowledge rules, i.e., rules that test the intrinsic characteristics of a block. In this example, the rules state that IF the block is black-and-white in color (i.e., after extracting the 'hue', 'intensity' and 'saturation' of the block by means of color filtering, IF we find that the 'saturation' value is zero), THEN the hypothesis that the block is a destination-address gets strengthened by the amount 0.25 .

Figure 3.2 shows a binary knowledge rule, i.e., one that tests the spatial relationships between different blocks. In this example, the rule states that IF a block B has been found to be a postage-stamp with a certainty factor X2, and if block A is to the left of and below block B, THEN the hypothesis that block A is a destination address gets strengthened by the amount 0.3 times X2.

Figure 4 shows a set of control rules. In this example, the rules state that to find the total certainty factor for the hypothesis that the given block is block type X, the system should execute all the 'isblock' rules for the block type X and then add up the 'confidence' values for all the rules that succeed. The sum thus obtained then gives the desired certainty factor.

Figure 5 shows a strategy rule. In this example, the rule represents the most straighforward strategy, i.e., to find a block of type X from among all the blocks in the image, the system should apply the control rules given in Figure 4 to all the data blocks in the image. The order in which the blocks are tested is determined by the ordered list *datablocks*, which is dynamically modified during the execution of the program and contains the block names in decreasing order of the likelihood of the block being of type X.

The strategy rule shown here queries the user on what kind of block is to be identified, and prints out an intermediate result. This is a simplification of the actual process where inputs are all taken from files, and the outputs are composed and put into files.

5. RESULTS

The expert system described is implemented on a VAX-11/780 running UNIX. The rule-based system is written in Prolog, and uses low-level image analysis routines written in C as well as intermediate image processing routines written in Lisp. The rule-based system currently contains ninety three rules. Of these, fifty seven are knowledge rules, twenty five are control rules and the remaining eleven are strategy rules. The domain for testing has so far been that of postal mail-pieces. The system was used in trying to locate the destination address block on pieces of letter-mail. The location of all the other blocks on the envelope image was also done in the process. The usual block classifications in this domain include 'destination address, 'return address', 'postage stamp', 'markup label', 'other text', 'graphics', etc. A variety of standard and non-standard mail-pieces that cannot be handled by the U.S. Postal Service's current OCR machines, were digitized and used as inputs to the system. A reasonably high degree of accuracy (over 80%) was achieved in this domain. A sample of the kind of envelope images used in the testing of the program is given in Figure 6. In the example shown, the system successfully classified the destination address, and also identified all the other blocks correctly. For the envelope images used, the rule-based system achieved an "understanding" rate of approximately one image per second; efforts are under way to improve this rate so that the system can be more effectively used as a practical tool for document understanding.

Addition of new rules for a larger variety of domains has been done, and more rules are continually being incorporated into the system. Creating rules for a wider set of domains poses no specific problems except that some of the rules have to be made more general (e.g., instead of trying to find an "address block", the same rules now try to locate a "block containing an address",

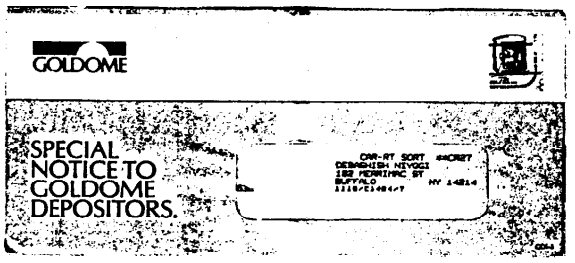

Fig. 6 Example of envelope image input for system

thus allowing for the existence of such a block in various kinds of documents). Further improvements are also being made to the handling of uncertainty, as mentioned above.

6. DISCUSSION

The wisdom of using production rules to represent knowledge has been the topic of discussion among many AI researchers. MYCIN, an early expert system, used production rules to encode knowledge about diseases caused by certain types of bacterial infection. Since then, many rule-based expert systems have been successfully developed for various domains. In the domain document understanding, a rule-based system is extremely elegant because unlike natural scenes, documents are very structured in character, and thus knowledge about features of documents can be very effectively formulated in terms of pro duction rules.

There are other advantages to using production rules in document image understanding. First, it is easy to apply either an additional strategy to a region that is hard to interpret with onnly one strategy, or a retry process having modified parameters. Second, software maintenance becomes easier, since addition/modification of rules is a relatively simple process that does not disrupt the rest of the system. Third, in a production system the processing is not carried out over the entire image uniformly, but only on necessary segments; thus, high efficiency is achieved. All these reasons make production rule-based systems eminently suitable for use in the domain of document under standing.

7. REFERENCES

[1] A.Barr and E.A.Feigenbaum, (ed.), *The Handbook of Artificial Intelligence, Vol. II*, William Kaufman Inc., 1982, 77-294.

[2] B.G.Buchanan and E.H.Shortliffe, *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, Massachusetts, 1984.

[3] W.F.Clocksin and C.S.Mellish, *Programming in Prolog*, Springer-Verlag, 1981.

[4] K. Kubota, O.Iwaki and H.Arakawa, "Document Understanding System", *7th international Conference on Pattern Recognition*, Montreal, Canada, July 30-Aug 2, 1984, 612-614.

[5] M.D.Levine, "A Knowledge-based Computer Vision System", in *Computer Vision Systems (Proceedings of a Workshop, Amherst, Massachusetts, June 1-3, 1977)*, A.R.Hanson and E.M.Riseman (ed.), Academic Press, New York, 1978, 335-352.

[6] D.M.McKeown, W.A.Harvey and J.McDermott, "Rule-Based Interpretation of Aerial Imagery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 5, Sept. 1985, 570-596.

[7] A.M.Nazif and M.D.Levine, "Low Level Image Segmentation: An Expert System", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, No. 5 (September 1984), 555-577.

[8] G.Nagy, S.C.Seth and S.D.Stoddard, "Document Analysis with an Expert System", *Proc. of Pattern Recognition in Practice II*, Amsterdam, June 19-21, 1985.

[9] S.N.Srihari, J.J.Hull, P.W.Palumbo, D.Niyogi and C-H Wang, "Address Recognition Techniques in Mail Sorting: Research Directions", *Tech. Report 85-09*, Dept. of Computer Science, SUNY at Buffalo, August 1985.

[10] S.N.Srihari, "Document Image Analysis", *unpublished manuscript*, Dept. of Computer Science, SUNY at Buffalo, February 1986.

[11] L.Wesley and A.Hanson, "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the Visions System", in *Proc. of the Workshop on Computer Visior* Ringe New Hampshire, Aug. 23-25, 1982, IEEE Computer Society Press.

[12] K.Y.Wong, R.G.Casey and F.M.Wahl, "Document Analysis System", *Proceedings of the 6th International Conference on Pattern Recognition*, Munich, Germany, Oct. 19-22, 1982.

[13] T.Weymouth, J.Griffin, A.Hanson and E.Riseman, "Rule Based Strategies for Image Interpretation", *Proceedings of AAAI-83*, August 1983.