What does a Machine Need to Know to Read a Document?

George Nagy
Rensselaer Polytechnic Institute
Troy, NY 12180

Abstract

The role of pragmatic, semantic, and syntactic knowledge in document reading and understanding is examined. Some sources of information to help interpret document layout, and common typesetting practices that facilitate OCR, are described.

Introduction

From an AI point of view, the types of knowledge required by a document processor are intermediate between *common-sense knowledge* and *expert knowledge*. In general, humans learn to understand complex documents (railroad schedules, catalogs, technical reports, even newspapers) only after they master the associated domain. We sketch out a taxonomy of the components of such knowledge and illustrate the actual and potential utilization of *knowledge bases* in document-image analysis and optical character recognition.

Most documents - books, journals, newspapers, letters - are organized hierarchically. (One might even argue that there can be no organization without hierarchy). A typical example of a *document hierarchy* might be volume, chapters, sections, paragraphs, sentences, words, letters. In spite of this organization, most documents are meant to be read in a specific linear traversal of the tree structure, called the *reading order*. (Works of reference are normally accessed through an index, but a reading machine must still read them from beginning to end, including the index. What is more troublesome are internal pointers to footnotes, figures, citations.)

Following Chomsky, we differentiate between the *deep structure* of a document and its *surface structure*. The conceptual content, called deep structure, is incorporated into the linear order as a string of concepts. These, in turn, are encoded into the surface structure of sentences, phrases and words. The atomic symbols of the surface structure are letters of an alphabet or ideographs. (At the risk of oversimplification, the difference between the two representations is that the deep structure should remain invariant when the document is translated from one language to another, while the surface structure changes.)

In the course of document production, the linear order is transformed into an ordered set of two-dimensional *pages*. While other physical forms exist - for instance, scrolls, and spiraling recitals of battles on obelisks - none matches the extraordinary convenience of print-on-paper. There is, however, no natural mapping from two dimensions to one. Therein lies much of the complexity of document analysis.

Over the centuries, the techniques of communication by means of graphic symbols on paper have been refined and standardized. *Layout* and *typesetting conventions* evolved for printed matter. Most such conventions provide guideposts for navigation in the tree (level indicators and node boundaries), but others give clues to emphasis, scope, dialog, citation, or exegesis.

The first task of document analysis is to recover from the printed page the writing symbols in the reading order that corresponds to the surface structure. The second spatial dimension does, however, provide many opportunities for encoding additional information and for speeding up the decoding process for the (human) reader. This information can be used, and must, in any case, be preserved, in the analysis.

Layout conventions are typically decided by an author or editor before a document is composed. They vary from publication to publication. For instance, whether the author's name precedes or follows the title is part of layout. Generic *typesetting conventions*, on the other hand, are of a consensual nature, and are violated only under extraordinary circumstances. Examples of typesetting conventions are the left-to-right order of the letters in English, and the alignment of the letters in a word on a baseline.

Understanding the layout and typesetting conventions not only helps in recovering the surface structure, but is also necessary for efficient extraction of the deep structure. Of course, every functioning document analysis and optical character recognition system has some layout and typesetting knowledge. What differentiates document analysis from optical character recognition is, however, the *extent* of such knowledge. The difference is comparable to that between a word-processing system and a desktop publishing system; in both cases, the boundary may be fuzzy, but the distinction is important.

In keeping with modern software-engineering principles, such knowledge should be codified and separated from the algorithms that actually perform the analysis. We hope that this survey will stimulate further progress in this important aspect of document analysis.

In the following sections, we discuss briefly methods for dealing with the deep structure and the surface structure of documents. Then we discuss in greater detail generic and publication-specific layout and typesetting conventions. Whenever possible, we illustrate the role of specific conventions by showing how document analysis systems can use them.

Deep structure

A universal document understanding system is probably still decades away: to facilitate the task, it is necessary to restrict the *domain of discourse*. For instance, in order to understand repair manuals for specific makes and years of automobiles, it would help the system to have access to a generic model of an automobile, just as a person would need some understanding of cars. The model could take the form of computer simulation, relational database, augmented transition network, frames and slots, scripts, or if-then rules. Document understanding would then result in the specialization of the generic model to a particular type of car.

A static *semantic database* for the selected domain of discourse could help resolve the ambiguity in the relationship of the modifier to the noun it modifies in syntactically equivalent phrases, such as *mineral oil* and *transmission oil*. Semantic analysis would also help the system to correctly interpret *the ?oints are burned*, where *?* might be either *j* or *p*.

A *pragmatic database* - dynamic, rather than static - could catch situation-dependent (causal) errors in interpretation. For example, it could understand a warning against *?isassembling the needle valve* after a sequence of instructions to take apart the carburetor for cleaning, where *?* might be *d* or *m*.

Pragmatic and semantic approaches to *discourse understanding* have long been the subject of study in artificial intelligence, linguistics, and information (library) science, but have not yet been brought to bear on reading machines. For many applications, it would be sufficient to have the document analysis system yield an output comparable to that of a word-processor. The resulting file can then be submitted to an automated indexer for subsequent information retrieval. Without full integration, however, the high-level domain-knowledge cannot be used to facilitate the pixel-level analysis.

Surface structure

The most immediate manifestation of surface structure is the language itself: English, French, Chinese. Knowledge of the language of the document being read is necessary not only to select the appropriate symbol alphabet, but also to restrict the vocabulary, syntax, and layout.

Syntactic conventions that prescribe word order are not absolute, but are generally honored in formal writing. *Parsers* for sentences and sentence fragments can be used to flag suspect words. *Markovian models* at the word level provide a simple alternative to formal parsing. Specialized syntactic conventions also dictate the placement of *punctuation marks* and the construction of *abbreviations, "written-out" numbers, numbers in numeric form, citations*, and *postal addresses*. These rules are listed in style guides ranging from a few pages for technical journals and mailing instructions to manuals of hundreds of pages. (An authoritative compendium on American usage is the *Government Printing Office Style Manual*).

Spell checkers based on word frequencies are less powerful but more commonly used in OCR than correction based on word sequences. A vocabulary 60,000-100,000 words provides very thorough coverage of English text, but three or four times more entries are needed for highly-inflected languages such as Italian or Hungarian. Storage of the lexicon is no longer a problem even on personal computers, but since every word in a document must be checked, fast access is essential. A simple trie or hash data structure is not adequate, because words with OCR errors must be retrieved.

The probability distribution of OCR errors differs from that of misspellings and common miskeyings (such as letter transposition) in word processing, and should be taken into account using a knowledge-base of *OCR confusion-pairs*. For instance, whereas the first letter of a word is seldom mistyped, recognition errors may occur anywhere, and *i-l* confusions are more common than in typing. Multiple *deletions* and *insertions* (due to improper character segmentations), *substitution* errors, and *wildcards*

(rejected characters) must be accommodated. Dictionary-organizations based on letter n-grams are less vulnerable to errors of this type.

In addition to error detection and correction, lexical analysis may also be used to improve the recognition process itself. For an incompletely identified word, candidates from the lexicon can be used to narrow down the search for the correct segmentation boundaries and character identities. Carrying this process to the limit, if similar patterns can be identified as representing the same symbol, then the recognition problem can be converted into a *substitution cipher* problem for which solution methods exist.

The lexicon can also be used to advantage in *training* the system to accept new typefaces by simply displaying alternative *word-hypotheses* during the training phase that the operator can confirm, key in, or reject. In addition to a standard lexicon, the user or the system itself may create customized word-lists, as is customary in standard word processors. These methods have been applied mainly to natural-language text, but sources of specialized vocabularies include *technical handbooks* (for example, names of chemical compounds); *telephone directories* for family names; *industrial and commercial directories* for corporate names; *country, state, city and street directories* for mail sorting; and *part-number catalogs*.

The morphology of words varies greatly from language to language. Most of this variation, such as the prevalence of words ending in "*ing*" in English or in "*e*" in French, is reflected in the statistical distribution of higher-order n-grams (including the blank as a special symbol). Indeed, the relative frequencies of a few characteristic bigrams and trigrams are sufficient to identify the language. (In speech, pairs or triples of phonemes can be used.) Error detection and correction methods based on *letter n-gram frequencies* are faster but less powerful than word-based methods. The two methods are sometimes combined to find the most probable word.

Certain constructs that appear in printed documents have their own specialized syntactic rules. Most of these are discipline-specific: mathematical formulas, chemical structure diagrams, arithmetic redundancy in financial documents, and so forth. Tables also often require more than casual attention for proper interpretation: consider, for instance, a calendar, the railroad timetables of a foreign country, or the IRS tax-tables.

Layout conventions

Most layout conventions are specific to a given family of documents. Items included in this category include the *page layout* (horizontal and vertical margins); the *column structure* (number and spacing); *paragraph indicators*; the placement, size, and types of *illustrations* and *tables* and their relation to *figure captions* and *labels*; and the disposition of non-narrative text.

In publications designed primarily to inform rather than impress, illustrations and tables are usually confined to rectangular frames. Pixel-level statistics allow discrimination between half-tones and line-drawings.

Printed forms and tables require specialized algorithms to process lines and boxes (*rules*). Most forms serve only an individual organization but some, like bank checks, are widely used with only minor variations.

In most publications, the column structure is relatively simple and uniform, but some newspapers and magazines vary the width and number of columns even on the same page. The location and orientation of the subtitles for columnar material depend, of course, on the direction (left or right, horizontal or vertical) of writing.

The positive identification of non-narrative logical entities, such as *headlines, headings, headers, footers, titles and subtitles, bylines, dates, mastheads, abstracts*, etc., etc., requires either a publication-specific knowledge base or powerful context processing based on OCR. An example of a set of layout rules for technical papers, prepared by Mary Guirsch and Junichi Kanai, appeared in the *Author's Kit Instructions* for this Symposium: incorporated into a knowledge base, it would presumably suffice for accurate spatial analysis of these *Proceedings*.

Publication-specific layout knowledge bases have been implemented using *expert system shells, if-then rules, geometric trees, form definition languages*, and *page grammars*. They have been demonstrated on newspapers, business letters, printed tables, resumes, technical journal articles and reports, trademarks, patent applications, typed forms with a specified layout, and even the periodical *Chess Informant*. (In reading chess, in addition to the syntax of chess notation, the semantics of the game were also used to great advantage.) Nevertheless, without automated methods, the development of publication-specific knowledge bases will remain a time-consuming task justifiable only for very high-volume applications.

Current OCR products provide files compatible with word-processors (*e.g. MS-Word, WordPerfect, WordStar*) or desk-top publishing systems (*e.g. TeX, TROFF, FrameMaker, Publisher, Interleaf*). What are potential target representations for the next generation of document readers? The separation of the layout structure from the logical structure meshes with document interchange standards such as the Open Document Architecture (*ODA*), Standard Generalized Markup Language (*SGML*) and Document Style Semantics and Specification Language (*DSSSL*). These can be visualized as a logical tree and a layout tree joined at the leaf nodes that consist of words.

In *ODA*, the logical structure consists of *chapters, sections, figures* and *paragraphs*. The layout structure is divided into *pages* and *blocks*. *Basic* and *composite blocks* form the *content architecture*. Sets of object classes and their relationships constitute the *specific structure* of a single document and the *generic structure* of a family of related documents; their characteristics are described by the *document profile* of layout and logical attributes. (We apologize for oversimplification: complete descriptions of *ODA* or *SGML* run to hundreds of pages.)

In order to use OCR in interpreting the layout, the symbol-processor must provide more information that just the symbol identities. Other desirable output includes the *alphabet* (Latin, Greek, Cyrillic, mathematical, phonetic, chess); *font family* (typeface and style - e.g. *Linotype Bodoni Book Italic*); *point-size; page coordinates; position with respect to the baseline* (drop-cap, subscript, superscript); *vertical and horizontal spacing; orientation; preceding and following symbols*. Current commercial OCR systems are *closed*: in the future, manufacturers may have to provide access, in the spirit of open systems, to internal parameters such as page-coordinates and recognition confidence for individual symbols, and permit use of external contextual filters.

Typesetting conventions

Except for the choice of typeface, typesize, and leading (between-line spacing) for various components of a document, typesetting conventions tend to be less publication-sensitive than layout conventions. A major reason for this is that originators of documents often relegate detailed typesetting to a human or automated compositor.

Typesetting rules are also built into most OCR systems intended for printed documents but, unlike layout conventions, they are seldom stored explicitly in a knowledge base. It is therefore often difficult to predict what configurations will give trouble. Further improvements in this area require more explicit codification of typesetting and even type-design conventions. Some examples of generic typesetting rules for text set in derivatives of the Latin alphabet are discussed below.

There are only a few ways in which typesetting can be used to demarcate paragraphs: the common methods - indentation, reverse indentation, line-to-line spacing, bullets - can be found in the style sheets of most word processors. (But the lead lines of paragraphs of European origin are often neither indented nor spaced.) The baselines within a paragraph are uniformly spaced. Paragraph identification tends to cause problems in OCR only when paragraphs are continued after a break (such as a multi-column illustration), or on the next column or page.

Printed lines are parallel and roughly horizontal, with each line set in a single point-size. Good typesetting practice dictates more than minimal (i.e., *unleaded*) separation between lines. The pronounced baseline of most typefaces (especially those with serif alignments) is intended to allow the reader to easily track individual lines. Automated systems are also able to exploit this feature. However, the presence of mathematical formulas, subscripts, superscripts, majescules and drop-caps may confuse simple-minded line-detection algorithms. (Entire lines of print are sometimes missed because of skew or geometric distortion introduced in the printing, copying, or scanning process, but this is not a typesetting problem.)

Word separation is difficult only when the difference between inter-character and inter-word spaces is obliterated by poorly executed justification. (L e t t e r s p a c i n g is sometimes used for emphasis in German.) The typeface never changes within the same word. Rules for hyphenation are gradually being relaxed, which may cause problems in lexical analysis. In some languages, underscores are used instead of line-break hyphens.

Character segmentation accounts for a large fraction of OCR errors and rejects. Among the major factors are poor print quality (speckle, drop-outs and character-fragmentation, particularly in copies of copies); digitization resolution inadequate for small type; kerning; italicization; boldface; very tight letterspacing; and unusual ligatures.

Once an individual character has been isolated, we must come to grips with the immense variability of typefaces. Several thousand are catalogued (classified by Maximilien Vox into eleven classes, adopted as a British Standard), with several hundred in wide use. Each typeface comes in a dozen or more *point-sizes* (which are *not* simply scaled versions of one another), *weights* (light, regular, bold, ultra) and *sets* (condensed, extended). A *font* of given typeface and point-size contains about 150 *sorts* (different slugs) in each of several *styles* or *variants* (roman, italic, small caps). The outlines of the

lettershapes of a given typeface vary in small detail from source to source (e.g. *Monotype, Compugraphic, Linotron*).

From an OCR perspective, the major divisions in typeface design are *proportional* vs. *fixed-pitch*, *serif* vs. *sans-serif*, and *body type* vs. *display type*. There are also typefaces (*OCR-A, OCR-B, Farrington 13-B*) designed specifically for ease of machine reading. Next to OCR fonts, fixed-pitch, sans-serif typefaces are easiest to segment, but they are unpleasant to read for humans.

For English, a sort normally consists of the upper-case and lower-case alphabets, *ligatures or logotypes* (*fi, fl, ffi, ft,...*), *figures* (numerals), *fractions*, *punctuation*, and *special symbols* (*&, $, @,...*). The width of the strokes may vary in a 4:1 ratio or more between the hairline bar in *e* and the stem of a *T*. The variation in stroke-width is even greater in printed ideographs that imitate brush strokes. In the English alphabet only *i* and *j* have more than one connected component, but in foreign words *diacritical marks* are commonly used.

The font is seldom changed within the body of a document, and even changes of style are relatively rare. It is therefore possible to take advantage of the similarity of bit-patterns that represent the same symbol in an entire stream of text to help identify noisy or garbled characters. Typeface homogeneity can be exploited not only for *reject recovery*, but also for *unsupervised learning* to adjust the parameters of the classifier.

Within a single typeface, there is always sufficient distinction between the patterns that represent different symbols. Some characters, such as *a, A, g, G, Q, 4*, and *&* vary a great deal from typeface to typeface and must be treated as different (but non-competing) subclasses in multifont classification. The variability can only be expected to increase as type-design tools are brought within the grasp of every computer user.

The pattern that represents letter *O* in one typeface may be identical to that for numeral *0* in another. A notorious triple is *I-l-1*. In some typefaces certain upper-lower case distinctions can be made only by size or position. It is therefore not always possible, even in principle, to classify single character-patterns in isolation.

Until recently a library of bit-patterns for many typefaces could be assembled only by laborious collection and scanning of sample alphabets. Now, however, hundreds of typefaces in many sizes are commercially available in digital formats that can be converted to bitmaps. Programs have also been developed that simulate the imperfections introduced by the printing and scanning processes. Over twenty different distortions and types of noise have been modeled.

So called *exotic* typefaces (non-Latin alphabets, Group XI in the Vox classification) require significant additions to the knowledge base. Chinese characters may have up to 30 strokes. Japanese page layout must accommodate interspersed Hurigana (explanations of the pronunciation of certain Kanji characters). In some languages such as Bengali, entire words may be linked, and the shape of a character in Arabic depends on those preceding or following it (like ligatures in English). Of course, syntax and morphology varies greatly even among Latin languages.

Conclusion

Some researchers consider character recognition only an experimental domain for the development of improved pattern classification algorithms. For this purpose, it is

sufficient to consider a set of isolated characters divided into a *training set* for deriving the parameters of the classifier, and a *test set* to provide a statistically credible measure of performance.

However, to build reading machines for converting paper documents into a useable computer-readable form, much information other than the shapes of individual characters must be taken into account. In principle, this information too could be learned from a training set of sufficient size and variety, but much of it already exists and awaits systematic integration into advanced document readers. Productive research on complete document-reading systems will be feasible for individual investigators or small groups only if diverse knowledge bases, OCR and document analysis software, and large samples of digitized and labeled document sets, become widely accessible.

Bibliography

ACM, *Proceedings of the ACM Conference on Document Processing Systems*, Santa Fe, ACM Order Number 429882, December 1988.

American National Standards Institute, *Office Document Architecture and Interchange Format*, 1988.

American National Standards Institute, *Information Processing - Text Composition - Document Style Semantics and Specification Language*, 1989.

H.S. Baird, *Document image defect models*, Pre-proceedings of the IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ, June 1990, pp. 78-87.

H.S. Baird, K. Thompson, *Reading Chess*, IEEE-PAMI 12, 6, pp. 552-559, June 1990.

H.S. Baird (editor), *Pre-proceedings of the IAPR Workshop on Syntactic and Structural Pattern Recognition*, Murray Hill, NJ, June 1990.

G. Boccignone, L. Freina, S. Mogliotti, M.R. Spada, *Towards an evaluation of an experimental OCR System by means of a complex document*, Progress in Image Analysis and Processing (V. Cantoni, L. Cordella, S. Levialdi, S. Sanniti di Baja, editors), World Scientific, pp. 543-550, 1990.

A. Bonnet, *Artificial Intelligence*, Prentice-Hall 1985.

A. Brown, *Type renaissance: A primer on digital type*, MacWorld, pp. 203-209, July 1991.

R.G. Casey and K.Y. Wong, *Document-analysis systems and techniques*, in Image Analysis Applications (R. Kasturi, and M.M. Trivedi, editors) Marcel Dekker, New York, 1990, pp. 1-36.

G. Ciardiello, M.T. Degrandi, M.P. Roccotelli, G. Scafuro, M.R. Spada, *An experimental system for office documents handling and text recognition*, ICPR-9, 1988.

Commission of the European Communities, *RIAO 91 (Recherche d'Information Assistee par Ordinateur) Conference Proceedings*, Barcelona, Spain, April 1991.

A. Dengel, G. Barth, *Document description and analysis by cuts*, Proc. RIAO 88, MIT, Cambridge, pp. 940-952, March 1988.

H. Fujisawa, A. Hatakeyama, *Intelligent filing system with knowledge-base*, Hitachi Review 37, 5, pp. 323-328, 1988.

H. Fujisawa, H. Yashiro, J. Higashono, Y. Shima, Y. Nakono, T. Murakami, *Document analysis and decomposition method for multimedia contents retrieval*, Proc. Second Int. Symp. on Interoperable Information Systems, pp. 231-238, INTAP, Japan, 1988.

V. Garza at al., *OCR Product Comparison*, Infoworld, pp. 73-90, October 22, 1990.

C.F. Goldfarb, *The SGML Handbook*, Oxford University Press, 1988.

L. Grunin, *OCR Software moves into the mainstream*, PC Magazine, pp. 299-350, October 30, 1990.

S.I. Hayakawa, *Language in Action*, Harcourt Brace, 1941.

H.S. Hou, *Digital Document Processing*, Wiley, 1983.

J.J. Hull, S.N. Srihari, R. Choudhari, *An integrated algorithm for text recognition: Comparison with a cascaded algorithm*, IEEE PAMI-5, 4, pp. 384-395, July 1983.

IAPR-IEEE, *Proceedings of the International Conference on Pattern Recognition* 1-10, 1973-1990.

International Typeface Corporation, *Upper and Lower Case, the International Journal of Typographics*, Vol 7-10, 1980-1983.

J. Johnson and R.J. Beach, *Styles in document editing systems*, Computer (IEEE), pp. 32-43, January 1988.

S. Kahan, T. Pavlidis, and H.S. Baird, *On the recognition of printed characters of any font and size*, IEEE-PAMI 9, 2, pp. 274-288, 1987.

Kanai, J., *Text line extraction and baseline direction,* Proc. RIAO 91, Barcelona, pp.193-210, Commission of the European Communities, 1991.

R. Kasturi, L. O'Gorman (guest editors), Special issue of *Machine Vision and Applications* on Document Image Analysis Techniques, to appear, Summer 1992

R. Kasturi, L. O'Gorman (guest editors), Special issue of IEEE *Computer* on Document Image Analysis Systems, to appear, June 1992

D.E. Knuth, *TeX and Metafont*, Digital Press and the American Mathematical Society, 1979.

D.E. Knuth, T. Larrabee, P.M. Roberts, *Mathematical Writing*, Stanford University Department of Computer Science Report STAN-CS-88-1193, 1988.

G. Lorette, C.Y. Suen (conference co-chairs), *Proceedings of ICDAR-91* (Int'l Conf. on Document Analysis and Recognition), St. Malo, France, September 1991.

Merghenthaler Linotype Company, *Linotype One-Line Specimens*, New York, 1958.

M. Minsky (editor), *Semantic Information Processing*, MIT Press, Cambridge, 1968.

R. McLean, *The Thames and Hudson Manual of Typography*. Thames and Hudson, London, 1980.

G.A. Miller, *Language and Communication*, McGraw-Hill, 1951.

R. Mohr, T. Pavlidis, A. Sanfelieu (editors), *Structural Pattern Analysis, Proceedings of the IAPR Workshop on Syntactical and Structural Pattern Recognition*, Pont-a-Mousson, France, World Scientific, 1989.

M. Nadler, *Document Segmentation and Coding Techniques*, CGVIP-28, 2, pp. 240-262, Nov. 1984.

C. Paoli and M.T. Pareschi, *A system for the automatic reading of printed documents*, Office Information Systems: The Design Process (B. Pernici and A.A. Verrijn-Stuart, eds), pp. 311-322, Elsevier 1989.

T. Pavlidis, S. Mori (guest editors), *Special issue of the IEEE Proceedings on Optical Character Recognition and Document Analysis*, to appear, Spring 1992.

E. Rich, K. Knight, *Artificial Intelligence*, McGraw-Hill 1991.

D.O. Robinson, M. Abbamonte, S.H. Evans, *Why serifs are important: the perception of small print,* Visible Language V, 4, The Cleveland Museum of Art, Autumn 1971.

M.G. Sabourin, A. Mitiche, *Optical character recognition by a neural network*, Proc. Neuro-Nimes, pp. 135-148, Nimes, November 1991, to appear in revised form in J. Neural Networks.

R.G. Schank and K.M. Colby (editors), *Computer models of thought and language*, W.H.Freeman, San Francisco, 1973.

H.R. Schantz, *The history of OCR*, Recognition Technologies Users Association, 1972.

S. Shlien, *Multifont character recognition for typeset documents*, Int. J. Pattern Recognition and Machine Intelligence 2, 4, pp. 603-620, 1988.

R.M.K. Sinha, *On partitioning dictionary for visual text recognition*, Pattern Recognition 23, 5, pp. 497-500, 1990.

R.M.K. Sinha, B. Prasada, *Visual text recognition through contextual processing*, Pattern Recognition 21, 5, pp. 463-479, 1988.

S.N. Srihari, *The Viterbi Algorithm*, Encyclopedia of Artificial Intelligence, Wiley 1986.

C.Y. Suen (editor), *Proceedings of the International Workshop on Handwriting Recognition*, CENPARMI, Montreal, April 1990.

C.Y. Suen, *n-Gram Statistics for Natural Language Understanding and Text Processing,* IEEE PAMI 1, 2, pp. 164-172, April 1979.

J. C. Vliet, *Proceedings of the International Conference on Electronic Publishing, Document Manipulation, and Typography*, Cambridge University Press, 1988.

H. Yashiro, T. Murakami, Y. Shima, Y. Nakono, H. Fujisawa, *A new method of document structure extraction using generic layout knowledge*, Int. Workshop on Industrial Applications of Machine Intelligence and Vision (MIV-89), Tokyo, pp. 282-287, 1989.