Toward a Taxonomy of Logical Document Structures

Kristen Summers
Department of Computer Science
Cornell University
Ithaca, NY 14853
summers@cs.cornell.edu

Abstract

The automated discovery of logical structure in text documents is an important problem that has recently received a good deal of attention; it can enable the creation of flexible and sophisticated document manipulation tools that will greatly increase the impact of electronic documents. This paper addresses aspects of the nature of the logical structures to be found, in order to develop categories of structures that reflect the variance in requirements for discovery and the variance in significance for applications. A complete taxonomy is not developed, but relevant attributes are identified in three forms of categorization: fundamental, based on structure definitions; discovery, based on required observables to find structures; and usage, based on roles structures play in applications. The attributes themselves are independent of the choice of particular logical structures to consider in a given application, and their direct implications are discussed.

1 Introduction

The automated discovery of logical structure in text documents is an important problem that has recently received a good deal of attention. A solution to this problem would, based on a representation of the physical instantiation of a document, create a hierarchy of the logical components of the document: paragraphs, sections, lists, etc. This hierarchy can enable a variety of applications in the realm of information access, including browsing, retrieval, and automated hyperlinking.[1] These applications can provide flexible and sophisticated document manipulation tools that will greatly increase the impact of electronic documents. This paper describes categorizations of logical components that will be useful both in designing solutions to the problem and in evaluating their performance.

1.1 Logical Document Structure

A logical structure tree for the present paper is given in Figure 1. (Other trees may be formed by including different degrees of granularity or organizing the components differently.) It should be clear that browsing may proceed based on tree navigation [4], hyperlinking may be performed by observing significant relationships between node values [1], a form of retrieval may be achieved by specifying tree locations of interest (and attributes they must have) [5, 13], the reuse of document portions will be eased by this kind of retrieval [15], and multiple style instantiations of the same document can be achieved by applying the corresponding style rules to a single tree [2, 8].

This logical structure should be distinguished from the *layout* structure that describes the physical text on the page and *content* structure that describes purely semantic relationships within documents, as follows.[2]

Definition 1 *The* logical structure *of a document consists of a hierarchy of segments of the document, each of which corresponds to a visually distinguished semantic component of the document.*

That is, logical structure lies at the intersection of content and layout; a logical segment must both be distinguished by its layout (thus the concept of a cohesive text passage is content only) and have meaning as a semantic unit (thus boldfaced text is layout

[1] When a marked-up version of the document is available (in SGML or another format), with a known set of markers, then the logical hierarchy is directly available. The problem of discovery arises when this is not the case, e.g., when a document is scanned-in, or when the available representation is in a Page Definition Language like PostScript.

[2] [12] recommends further distinguishing the *geometry* of a document, which includes line breaks, page breaks, etc., from the *layout*, which describes only the formatting guidelines, such as left-justification.

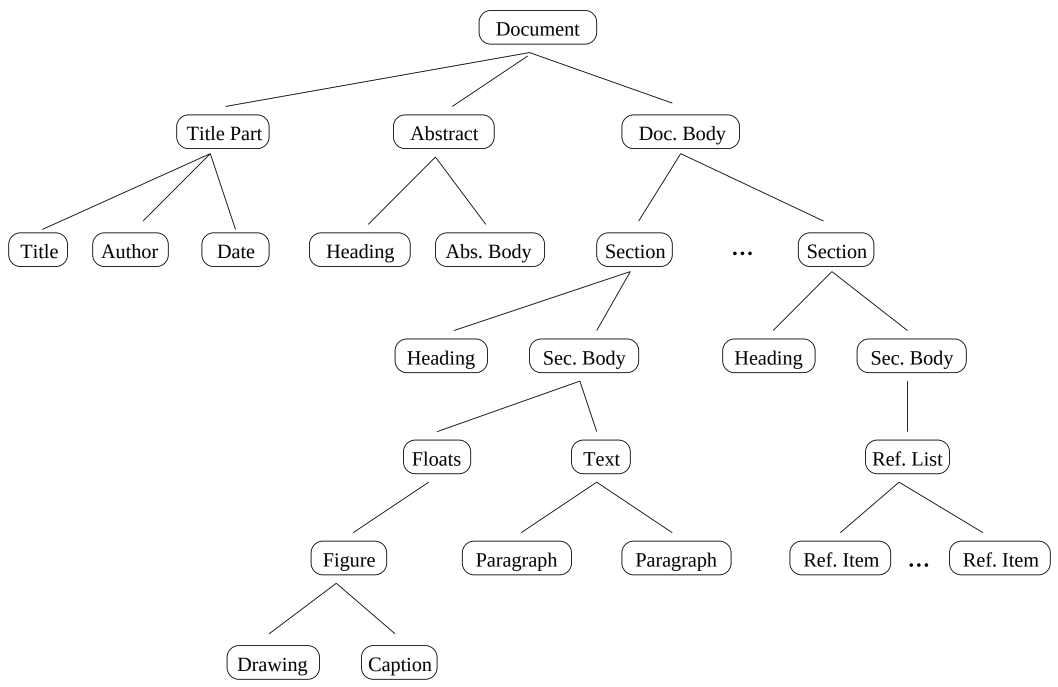

Figure 1: A logical structure tree for the present paper

only). Furthermore, the logical structure of a document refers to the hierarchy formed by the containment relationship among these components; other relationships exist, such as that formed by references in the text, but these do not form a part of the logical structure in the current sense.[3] Each type of element in such a hierarchy is also referred to as a logical structure.

1.2 Typing Logical Structures

Work in the area of logical structure discovery and use raises interesting questions about the nature of these structures. Can we find classes of structures that correspond to degrees of difficulty in their discovery, based on the necessary observables and/or a space-time complexity kind of analysis? Can we find classes of structures that correspond to their significance for the user? This paper does not provide a complete answer to these questions, but it does identify many attributes that will need to be considered in the attempt to find such an organization of logical document structures. These attributes and categorizations exist independently of the choice of particular structures to consider in a given application. The categorization of a particular logical structure does depend on its definition, which may vary in different contexts and applications; the examples in this paper are intended to follow sufficiently general rules as to be applicable in most settings.[4]

The following subsection discusses related work; the remaining sections discuss different forms of categorizations of logical document structures. These logical structures can be categorized in at least the following ways: fundamental distinctions, based on the definitions of the structures themselves and discussed in Section 2; discovery distinctions, based on the observable characteristics required (or helpful) for identifying the structures and discussed in Section 3; and usage distinctions, based on the kinds of use of the structures that can be expected in applications and discussed in Section 4.

[3] The approaches to logical structure discovery cited here focus on the problem of finding the logical structure in this sense (called the *primary structure* in [10]); other relationships can often be derived directly from a combination of this tree, effective optical character recognition, and a small amount of text analysis.

[4] For instance, an assumption in this paper is that the structures theorem and definition should be distinguished but are not formatted differently according to any predefined rule; there may be contexts in which they are formatted differently and this is known *a priori*, but this is not typical.

1.3 Related Work

Much work in the area of logical document structure focuses either on its automatic discovery or on application of the information contained in this structure, and some work addresses the question of how these structures should be represented.

Discovery Approaches to logical structure discovery assume some prior knowledge of the style of the document, i.e., the effects of logical structure on layout; layout observations are then analyzed to determine the logical structures that caused them. The required information about the document style ranges from very specific and precise to fairly general ideas about the ways in which it is possible to convey logical information through formatting. This information is presented as a grammar, and the document layout is parsed in [3, 9, 11, 12]. Other approaches of varying degrees of similarity to parsing and based on varying degrees of knowledge specificity, are presented in [6, 7, 16, 18, 21, 22, 23, 24, 25].

Applications Applications of the solution (which may be represented as a separate hierarchy, with pointers to document locations, or as a marked-up version of the document) are discussed in [1, 2, 4, 5, 13, 15, 19, 20, 26]. These applications include, but are not limited to, those discussed in Subsection 1.1.

Representation Discussions of types of logical structure representation can be found in [10, 17]. [10] provides a taxonomy of full structural hierarchies, considering attributes such as the choice of atomic unit of structure. [17] formalizes and extends ODA (Office Document Architecture) in an object-oriented framework; in the process, an object taxonomy is presented that distinguishes between layout and logical objects and between simple and composite objects. Further distinctions among logical object classes are not explored, as they do not affect the goal of the paper.

This paper differs from the above work in that its concern is with the nature of the units of logical structure themselves. Since the attributes of logical structures have implications for the kind of work described above, it is to be hoped that the current interest in finding and using logical structures will lead to more complete explorations of the nature of these structures.

2 Fundamental Divisions

The most basic divisions of logical structures rest on the definitions of the structures themselves. These distinctions have obvious, direct implications for the structure discovery; what is included in the definition of a structure affects the preferred method of identifying this structure.[5] They also affect the other categorizations. These divisions include distinctions between primary and secondary structures and between content-oriented and layout-oriented structures.

2.1 Primary vs. Secondary

Primary structures are defined, at least in part by their own attributes; *secondary* structures can be completely defined by their positions in the hierarchy and relationships to other structures. For example, a section heading is a primary structure; it is identifiable by its appearance and separation from the surrounding text. This primary structure provides the basis for finding the secondary structures section body and section. A section body is a right sibling of a section heading with, in turn, no right sibling of its own;[6] a section is a node whose children are exactly a section heading and a section body. Figure 2 shows Figure 1, with primary structures in solid boxes and secondary structures in dashed boxes.

2.2 Content- vs. Layout-Orientation

Another fundamental distinction can be made based on the relative roles of content and layout in the definition of a logical structure. Although both must be included, some logical structures can be considered *content-oriented*, and some can be considered *layout-oriented*. For example, a definition is a logical structure when it is distinguished by its presentation, as in Section 1 of this paper; it remains, however, a content-oriented structure. On the other hand, a special paragraph (a paragraph presented in other than the usual format for a given document[7]) is a layout-oriented structure. These descriptions are relative; a logical structure is more content-oriented than another if its definition relies more heavily on internal

[5] The structure definition does not completely determine how to discover it, however; extra-definitional cues may be quite useful, and at times it may be appropriate to categorize a document piece as a structure whose definition it does not match precisely.

[6] This definition refers to an ideal tree, in which the sections have been correctly identified. In the process of forming a tree with an imperfect method, a more useful definition might be: a section body is a right sibling of a section heading whose own right sibling, if it exists, is also a section heading.

[7] In this paper, definition is a subtype of special paragraph.

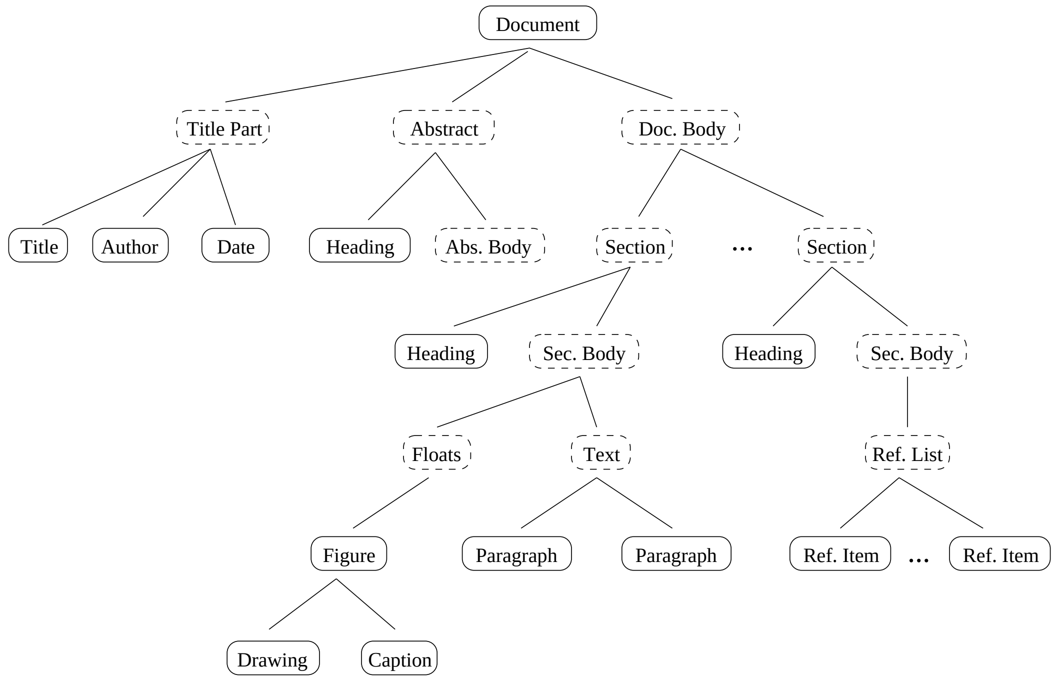

Figure 2: The earlier tree, with primary and secondary structures distinguished

meaning; similarly, a more layout-oriented structure has a definition that relies more heavily on visual presentation.

To make this precise, consider the hierarchy that can be formed among logical structures themselves, in which the children of a node are subtypes of the node's structure.[8] A portion of this hierarchy is given in Figure 3. If a structure is distinguished from its siblings entirely by content, it is content-oriented; if it is distinguished from its siblings entirely by layout, it is layout-oriented; otherwise, it is neither.

Degrees of this kind of orientation are distinguished by the degree to which this definition can be extended. That is, a structure that is distinguished from its siblings and its parent's siblings by content alone is more content-oriented than one that is distinguished from its own siblings by content alone but from one or more of its parent's siblings in part by layout. (Note that if a node is distinguished from its parent's siblings by content alone, it is therefore also distinguished from its first cousins, i.e., its parent's siblings' children, by content alone.) A structure that is also distinguished from its grandparent's siblings by content alone is more content-oriented still, etc. This is equivalent to the idea that the degree of content-orientation corresponds to the number of immediate ancestors of a content-oriented structure that are also content-oriented. Of course, this definition of degrees of orientation also applies analogously to layout-orientation.

[8] Typically, the logical structure of a document will be given in terms of the structures at the leaves of this hierarchy.

3 Discovery Divisions

Primary logical structures can be characterized by the cues that are necessary and/or useful in their discovery. (Secondary structures can, of course, be discovered by applying their definitions after primary structures have been found; thus, no additional cues are needed.) These cues belong to four basic categories: *geometric*, *marking*, *linguistic*, and *contextual*.

Table 1 at the end of this section provides several examples of necessary and useful discovery cues for primary logical structures, in terms of these categories. Necessary cues are marked as "Nec.," and useful cues are marked as "Helps." Note that a structure must require at least the observables required by its ancestors in the structure hierarchy of Figure 3.

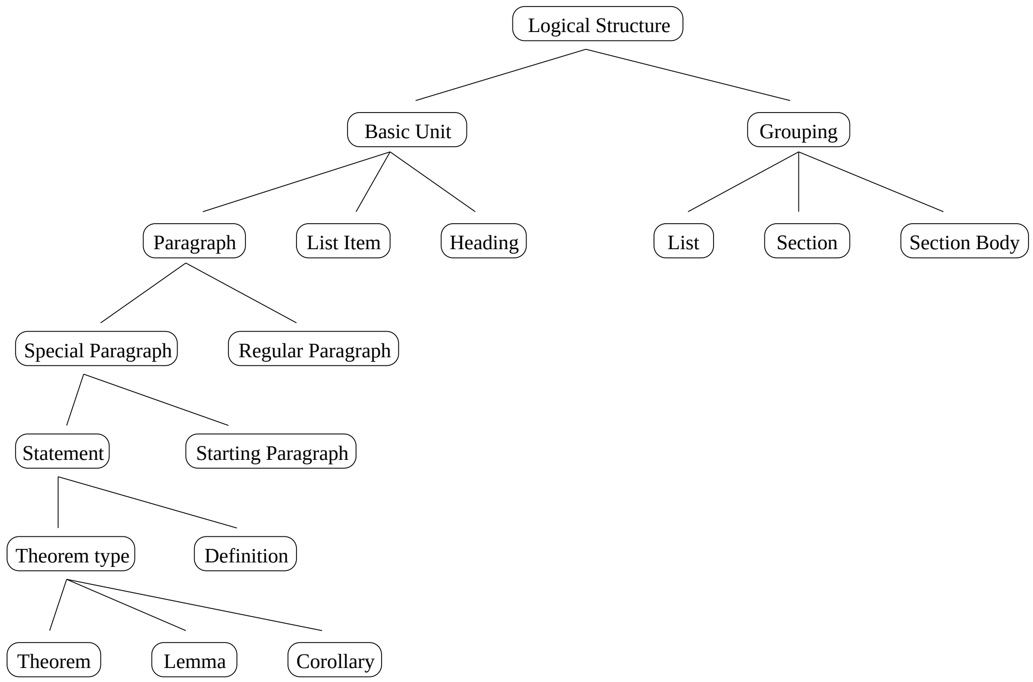

Figure 3: A partial hierarchy of logical structures

3.1 Geometric Observables

Geometric observables include the (external) *contours* and the *internal shape* of a piece of text. (*Height* is a special case of *contours*.) Both of these kinds of cues may be necessary; for instance, the *contours* of an indented list provide its characteristic shape of a hanging indent, but the shape of a table is recognized by the *internal shape* of its columnization [14]. Since geometry involves the shapes formed by the marks on the paper or screen, its contribution can (inversely) be found by an analysis of the white space in a document [21].

3.2 Marking Observables

Marking observables consist of non-linguistic marks on the paper or screen; this includes attributes like font type and weight, as well as non-alphanumeric *symbols*, such as bullet points and rule lines. Bullet points and dashes, for instance, can aid in the identification of indented list items; symbols can be necessary to find left-justified list items.

For example, consider Figure 4, in which a portion of an actual e-mail message is represented with different sets of observables. In all cases, alphabetic characters have been replaced with the letter "x." In the upper left version, all symbols and characters have been so replaced; in this representation, no difference in the format of the text blocks is visible. In the upper right version, the observables include *symbols*; the lower two text blocks can be observed to begin with a parenthesized character, suggesting that they are items in a left-justified list. In fact, this is so, as is quite clear in the lower right version, in which both *symbols* and numbers are included. (In this case, either *symbols* or numbers are sufficient to suggest the presence of a list without the other, but either may be required, depending on the form of marking the list items.)

3.3 Linguistic Observables

Linguistic observables include combinations of *numeric* and *alphabetic* symbols. (These cues enter a gray area between *symbolic* and *linguistic* when they are character-based rather than word-based.) The observation of words is necessary for structures such as theorem to be recognized and distinguished from similar structures (e.g., definitions, in many cases). The identification of indented list items is aided by *numeric* cues just as by *symbolic* ones; again, these cues can be necessary to identify justified list items. For example, if the item numbers in Figure 4 were

Figure 4: Paragraphs and a justified list, with and without observable symbols and numbers

not enclosed in parentheses, *symbols* would not identify the list, and *numeric* cues would be necessary to find it. In its current form, *numeric* cues are enough to suggest that it is a list, as can be observed from the lower left representation.

Typically, content-oriented structures will require linguistic cues, since content is usually contained in the language of a document. This linguistic analysis can remain quite shallow or become very complex; naturally, the subtlety of the content aspects of discoverable structures depends in part on the depth of the analysis. For example, consider an attempt to distinguish an author's institutional affiliation (one structure) from address (another structure), without making use of further analysis than checking for the presence of keywords. In order to find most institutions, the relevant keywords would probably include "University, College, Corporation, Company," to name but a few; the effect would be that streets like "University Avenue" and towns like "College Park" would be incorrectly identified as affiliations.[9] The goal is probably not reasonable for the amount of included analysis.

3.4 Contextual Observables

Contextual observables can be divided into *local* and *global* context-based cues. Local contexts use information about some limited number of surrounding nodes: siblings, parents, children, or neighbors within a level (which may or may not be siblings). For instance, consider a typical business letter; the return address and the closing (including the signature, etc.) are both internally left-justified blocks indented approximately halfway across the page; in this setting, they can be distinguished easily by local context, since the return address is not preceded by any text, but the closing is.

Global contexts use information about the document as a whole. For example, a special paragraph is a paragraph that differs in its presentation from the typical paragraphs within the document.[10]

Contextual information may, of course, include information of any of the preceding varieties; moreover, it may make use of available structure type information.

[9] Some cases could be filtered out by requiring that affiliations not contain numeric values, but this would not cover every case.

[10] Identifying the typical is a significant problem, as the standard may not always occur more frequently than the non-standard.

4 Usage Divisions

The logical structures of a document may also be characterized according to their use. This kind of categorization attempts to capture information about the relative significance of different logical structures; it has implications for performance evaluation of logical structure discovery.

The relative importance of logical structures is, of course, application-specific. As an extreme example, consider the application of a theorem extractor. For this tool, theorem is the only structure of direct significance. Its ancestors in the structure hierarchy are also important, to the extent that errors in their identification may lead to errors in identifying theorems; no other structures matter, so errors in their identification are insignificant. Finding the full logical structure would be overkill for this application, but the point stands that if a structure discovery mechanism is designed for a particular application, its output should be evaluated with respect to structure importance within that application.

In the more general case, however, the logical structure is derived for possible use with many applications, and the kind of information described above is therefore unavailable. A more general (and necessarily less precise) concept of structure significance is required. This raises a variety of different issues, including: classifier implications, expected user references, hierarchy role, and generality. These are described below.

Table 2 at the end of this section provides several examples of the usage characteristics of logical structures, in terms of these categories.

4.1 Classifier Implications

This attribute refers to the relevance of a structure to the identification of other structures. To some extent, this is dependent upon the classifier itself, but it also depends on the intrinsic definitions of the structures; the definitions of secondary structures highlight the importance of certain other structures, as do the definitions of primary structures that depend in part on their contexts (e.g., special paragraph relies on other paragraphs). For example, section headings are quite significant in this respect, as two secondary structures (section bodies and sections) rely on them for correct identification.[11]

[11] Here, as elsewhere in this paper, the word "section" may be replaced by "subsection" or "sub^{n}section" where $n \geq 0$.

Table 1: Some Primary Structures and Discovery Cues

Structure	Geometry		Marking		Linguistic		Context	
	Contour	**Internal**	**Font**	**Symbol**	**Word**	**Number**	**Global**	**Local**
Paragraph	Nec.	Helps		Helps				
Special Paragraph	Nec.	Helps	Helps	Helps			Nec.	Helps
Theorem	Nec.	Helps	Helps	Helps	Nec.		Nec.	Helps
Indented List Item	Nec.		Helps	Helps		Helps		Helps
Justified List Item	Nec.		Helps	Nec.		Nec.		Helps
Indented List	Helps							Nec.

4.2 Expected User References

Structures that users refer to more frequently than others are, in an important sense, particularly significant. For example, if users write queries that ask for full sections more often than groups of paragraphs, then sections are more significant for retrieval than are paragraphs (unless, of course, these sections are frequently identified by particular paragraphs they contain).

This attribute is task-dependent; different structures may be commonly used in retrieval from those commonly used in browsing, for example.

Determining precise expectations for this would require tracking the behavior of actual users with a fully general system (including different kinds of documents). Such a study would provide a solid basis for according relative weights to different logical structures, with respect to this attribute.

In the absence of strong empirical evidence, however, certain general observations can be made, based on the natures of commonly suggested applications.

- In hierarchical browsing as described in [4, 21], structures at higher levels of the tree are more significant than those at lower levels. Since this browsing is based on tree navigation, starting from the root, higher-level structures will be used for earlier decisions, on which later decisions will in part rely. Furthermore, for any node that is accessed, all of its ancestors must have been accessed as well, but its descendents need not be. So for this application significance corresponds (to a large degree) to height.

- For hyperlinking, bibliographic structures have a special significance. Links will often be desirable based on bibliographic matches (such as articles that share authors, or a match between a reference in one article and the title of another), so the structures that provide this information are particularly important. Floats (figures, tables, etc.) have a similar importance for linking, as they are typically referenced in the text.

  More conceptual hyperlinks are often desirable as well, of course. It is not obvious how these will relate to logical structure, however, as the criteria for their inclusion are still emerging.

- In the retrieval of previously-seen documents or document portions, highly salient structures are likely to be those which differ greatly from their surroundings, thereby calling attention to themselves. (This is an intuition, which can be verified or disproved by tracking use of a system with logical structure-based retrieval. Furthermore, determining which structures differ greatly from their surroundings in this sense is far from trivial.)

The above is not meant to be exhaustive; it simply provides an example of the kinds of issues that can provide insight into the significance of different logical structures from the direct perspective of a user.

4.3 Hierarchy Role

A significant distinction can be drawn between those structures that exist in order to express a useful piece of the document and those that exist in order to complete the hierarchy. For example, the structure paragraph part is not, in itself, useful; it exists in order to complete the children of a paragraph that contains an equation or an indented list or some other interruption. *Filler* structures that exist only to complete the hierarchy are a proper subset of secondary structures. Useful structures are, of course, more significant than are fillers (although distinguishing the two may be extremely significant!)

Table 2: Some Structures and Usage Characteristics

Structure	Implications	Task Importance	Hierarchy Role	Generality
Paragraph	Para. Group		Useful	High
Heading	Section Body, Section	Browsing	Useful	High
Section		Browsing	Useful	High
Section Body		Browsing	Filler	High
Title Part	Title, Author, etc.	Browsing	Useful	High
Title		Browsing, Linking	Useful	High
Corollary	Proof		Useful	Low

4.4 Generality

Consider the structure hierarchy, partially shown in Figure 3. Distinguishing between structures that appear at lower levels of the tree changes the meaning of the result less than distinguishing between structures at higher levels; thus those at higher levels are more significant, in that their correct identification provides more new content.

5 Conclusions

This paper has described several criteria for categorizing the types of logical structures of text documents. Although a general system for typing these structures has not been achieved, many issues have been raised that must be considered in the process. These fall into three (sometimes interrelated) categories: fundamental distinctions, based on structure definitions; discovery distinctions, based on necessary and useful cues for structure identification; and usage distinctions, based on structure roles in applications. Identifying these attributes and differences provides an important step towards developing a more general theory of classes of logical document structures. This problem deserves further attention, as its solution will have significant implications for the development and evaluation of logical structure discovery techniques.

Acknowledgments

I am very grateful to John Hopcroft for proposing the problem and for his guidance and support; I am also very grateful to Daniela Rus for her guidance. Special thanks to Jim Davis for a careful reading of an early draft and to and all members of the Information Capture and Access group at Cornell for enlightening discussions. Thanks also to the anonymous reviewers for many helpful comments.

References

[1] James Allan, Jim Davis, Dean Krafft, Daniela Rus, and Devika Subramanian. Information agents for building hyperlinks, 1993.

[2] Dennis S. Arnon. Scrimshaw: A language for document queries and transformations. *Electronic Publishing*, 6(4):385–396, 1993.

[3] Henry S. Baird. Anatomy of a versatile page reader. *Proceedings of the IEEE*, 80(7):1059–1065, 1992.

[4] Victoria A. Burrill. VORTEXT: VictORias TEXT reading and authoring system. In J. C. van Vliet, editor, *Text Processing and Document Manipulation: Proceedings of the International Conference*, British Computer Society Workshop Series, pages 43–57, Nottingham, April 1986. Cambridge University Press.

[5] Charles L. A. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation, August 1994. URL: `ftp://cs-archive.uwaterloo.ca/cs-archive/CS-94-30/structxt.dvi`.

[6] Denise Derrien and Michel Habib. Approche objet pour l'analyse de la structure logique des documents. In Jacques André and Jean Bézivin, editors, *Woodman '89: Workshop on Object-Oriented Document Manipulation*, pages 226–235, Rennes, May 1989.

[7] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. Multistrategy learning for document recognition. *Applied Artificial Intelligence*, 8:33–84, 1994.

[8] An Feng and Toshiro Wakayama. SIMON: A grammar based transformation system for structured documents. *Electronic Publishing*, 6(4), 1993.

[9] Hiromichi Fujisawa, Yasuaki Nakano, and Kiyomichi Kurino. Segmentation methods for character recognition: From segmentation to document structure analysis. In *Proceedings of the IEEE*, volume 80, pages 1079–1092, July 1992.

[10] Richard Furuta. An object-based taxonomy for abstract structure in document models. *The Computer Journal*, 32(6):494–504, 1989.

[11] Tao Hu and Rolf Ingold. A mixed approach toward an efficient logical structure recognition from document images. *Electronic Publishing*, 6(4):457–468, 1993.

[12] Rolf Ingold. Text structure recognition in optical reading. In Jacques André, Richard Furuta, and Vincent Quint, editors, *Structured Documents*, The Cambridge Series on Electronic Publishing, pages 133–141. Cambridge University Press, Cambridge, 1989.

[13] Michael H. Kay. Textmaster – document filing and retrieval using ODA. In J. C. van Vliet, editor, *Text Processing and Document Manipulation: Proceedings of the International Conference*, British Computer Society Workshop Series, pages 125–139, Nottingham, April 1986. Cambridge University Press.

[14] Christopher Lewis, Daniela Rus, and Matthew Scott. A structure detector for tables. Forthcoming Technical Report.

[15] Keith McAlpine and Paul Golder. A new architecture for a collaborative authoring system: Collaborwriter. *Computer Supported Cooperative Work*, 2:159–174, 1994.

[16] Masaaki Mizuno, Yoshitake Tsuji, Toshiyuki Tanaka, Haruhiko Tanaka, Masao Iwashita, and Tsutomu Temma. Document recognition system with layout structure generator. *NEC Research and Development*, 32(2):430–437, July 1991.

[17] Makoto Murata. An object-oriented interpretation of ODA. In Jacques André and Jean Bézivin, editors, *Woodman '89: Workshop on Object-Oriented Document Manipulation*, pages 91–100, Rennes, May 1989.

[18] Gilbert B. Porter and Emil V. Rainero. Document reconstruction: A system for recovering document structure from layout. In *Proceedings of the Conference on Electronic Publishing*, pages 127–141, 1992.

[19] T. V. Raman. *Audio System for Technical Readings*. PhD thesis, Cornell University, May 1994. URL: `http://www.cs.cornell.edu/Info/People/raman/phd-thesis/`.

[20] Daniela Rus and Devika Subramanian. Multimedia RISSC Informatics: Retrieving Information with Simple Structural Components. In *Proceedings of the ACM Conference on Information and Knowledge Management*, Washington DC, November 1993.

[21] Daniela Rus and Kristen Summers. Using white space for automated document structuring. In *Proceedings of the Workshop on Principles of Document Processing*, Seeheim, 1994.

[22] Yasuo Tanosaki, Kenji Suzuki, Kiyoshi Kikuchi, and Motoshi Kurihara. A logical structure analysis system for documents. In *Proceedings of the Second International Syposium on Interoperable Information Systems*, pages 221–228, Tokyo, November 1988.

[23] Dacheng Wang and Sargur N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47:327–352, 1989.

[24] Toyohide Watanabe, Qin Luo, and Noboru Sugie. Structure recognition methods for various types of documents. *Machine Vision and Applications*, 6:163–176, 1993.

[25] K. Y. Wong, R.G. Casey, and F. M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, November 1982.

[26] Haviland Wright. SGML frees information. *Byte*, 17, June 1992.