Gary Perlman & Thomas D. Erickson

Graphical Abstractions of Technical Documents

Good technical writing demands clear and concise communication that allows readers to skim documents for efficient access to information. To aid technical writers many computer programs have been written to analyze writing style in the hopes of improving writing standards. These programs have tended to be of a numerical statistical nature, summarizing a document or predicting its "goodness." We feel such programs hide more information than is advisable to help writers understand where and why their documents may have difficulties. After introducing the general concept of an abstraction of a document, we describe the other side of the text analysis coin: graphical displays of text that enhance structural components of a document. We describe two programs for graphical textual analysis: one generates displays of the logical structure of sections of a document; the other generates graphs of the complexity of individual sentences. While these programs are not the final statement of abstract text analysis, they point a new direction in which we think writing aids should be going.

1. Good Technical Writing

When we say a document should be well-written we really mean that it should be easy to read. This includes having well structured sections, smoothly flowing prose, and well written sentences. But well-written documentation is something more than just well-written prose.

People rarely sit down and read a document through. They usually approach a document wanting to know something in particular (Wright, 1983). They have a goal in mind. Some of a reader's more common goals are to:

Determine if the document contains the information being sought.
Determine if the document is of sufficient interest/use to be read more thoroughly.
Find information about a particular topic (e.g., how to use a program).
Use the document as a reminder of information once learned.

In addition to the information-oriented goals, there is probably the single most common and constraining of all the goals that readers bring to their task: Get this done in a hurry so I can do something important.

Documentation must be easy to understand not only when it is read, but also when it is skimmed. It is more important that technical documentation be skimmable than that it be readable. Sections should be used to separate distinct ideas, and their headings should give the reader an overview of a document. The first paragraph of a section should introduce the main idea of the section, and the first sentence of a paragraph should give the topic for the paragraph.

Visible Language, XVII 4 (Autumn 1983), pp. 380-389.
Gary Perlman's address: AT&T Bell Laboratories, Murray Hill, NJ 07974.

Both these devices help smooth the flow of ideas in a document and make it easier to get the gist of the document in a minimal amount of time. The key to making a document skimmable is to give it a rich internal structure.

At a more microscopic level it is important to have the parts of a document well written. Once readers are within a section of text, such as a paragraph, they must find the individual sentences easy to read. Sentences must be of reasonable length and of limited grammatical complexity.

In the rest of this document we briefly summarize some of the more traditional approaches to computer aids to good writing. We point out some of their deficiencies and suggest new directions. We conclude with two exemplary programs that graphically summarize document and sentence structure.

2. Traditional Aids: Summary/Predictive Statistics

Traditional document analysis primarily employs summary and predictive statistics. After submitting a document to an analysis program, an author may be given a table of statistics like the output of the style program (Cherry, 1980) shown in Figure 1.

Figure 1. Summary statistics for this document. The printout of the STYLE program for a draft of this document provides a large variety of numerical statistics. The readability grades are predictors of the number of years of formal education needed to understand the text and are based on integrating summary statistics like sentence length and word length.

```
Readability Grades:
   Kincaid 11.9 auto 12.4 Coleman-Liau 13.2 Flesch 14.0 (43.2)
Sentence Information:
   no. sent 150 no. wds 2798
   av sent leng 18.7 av word leng 5.19
   no. questions 2 no. imperatives 0
   no. content wds 1687 60.3% av leng 6.79
   short sent (<14) 29% (43) long sent (>29) 9% (13)
   long sent 69 wds at sent 70; short sent 3 wds at sent 114
Sentence Types:
   simple 51% (77) complex 29% (43)
   compound 10% (15) compound-complex 10% (15)
Word Usage:
   verb types as % of total verbs
   tobe 37% (112) aux 23% (70) inf 20% (59)
   passives as % of non-inf verbs 16% (39)
   types as % of total
   prep 10.8% (302) conj 2.7% (76) adv 3.6% (100)
   noun 29.2% (816) adj 19.0% (531) pron 4.3% (119)
   nominalizations 2 % (69)
Sentence Beginnings:
   subject opener: noun 22 pron 22 pos 1 adj 32 art 31 tot 72%
   prep 13% (20) adv 5% (8)
   verb 3% (4) sub_conj 2% (3) conj 1% (2)
   expletives 3% (5)
```

Many people do not understand most of these summary statistics and have trouble interpreting their significance. The difficulty with interpreting summary statistics is partly solved by the use of predictive statistics such as a readability index, an integration of a set of summary statistics shown to be statistically correlated with ease of reading (speed, comprehension, etc.).

There are problems with predictive statistics too. For example, although average sentence length is positively correlated with readability, this does not mean that a long sentence will necessarily be more difficult to understand than a short one. Often a longer complex sentence can express an idea more clearly than several short sentences, especially if relationships between ideas are presented. Predictive statistics ignore semantics and gloss over individual cases, making them of dubious validity and of questionable utility. We have observed inexperienced writers splitting up perfectly good sentences to make them shorter to get a better readability score. There are also problems for passages and for documents for which readability standards have not been set. These and other problems with statistics are summarized by Coke (1982).

Most statistics are global measures and as such offer little information about the source or solution of problems. Trouble areas need to be confined to a particular section of the paper, and the type of problem and its solution should be made easy to identify.

3. A More General Approach: Abstractions

Before outlining our approach to a solution to the problems of traditional text analysis programs, we will introduce our notion of text abstraction. An abstraction of a document is a summary of a part of it that focuses a writer's attention on a particular aspect of that document, for example, section structure or sentence complexity. An abstraction strips away irrelevant or redundant information which may hinder analysis. The traditional approach to text analysis (statistics) is a subset of the abstraction view. The main difference is that part of our generalization is the notion of a graphical summary display of a document.

A graphical display of some text has the property that some physical attribute of the display corresponds to some property of the text. This allows a person to see the logical structure of sections in a document or the complexity of a sentence. Graphical displays offer a richer source of information than numerical summaries. They almost literally demonstrate that a picture is worth 1000 words.

The generalization of traditional statistics on text to abstractions is analogous to the generalization of statistics to data analysis. We are treating text as a special type of data to be analyzed. Just as graphical displays of data offer more information about data in a way people often more readily understand, graphical displays of text can present a more clear and concise summary. And just as graphs of data are less judgmental than predictive statistics, so are graphs of text. They allow people to make their own conclusions based on more information than statistics alone.

4. Two Graphical Abstraction Programs

The programs described here run on the UNIX (trademark of AT&T Bell Laboratories) operating system (Richie & Thompson, 1978) and are designed for use with the troff text formatting system, although the PUNC program can be used with any UNIX text processing system. The programs are simple enough that they can be implemented on any system with minimal programmer effort.

We do not think these are ideal tools for textual analysis, but we do think they give a new direction for text analysis. Over the years they have been in use, people at our computer facility have found them useful. Experienced writers prefer them to the more traditional programs because the programs *help* with the analysis rather than *do* the analysis.

4.1 HEADINGS: Extract Section Headings

On our computer system at the University of California at San Diego (UCSD), we use a text processing system that prints documents in a format defined by a set of macros (text commands) that define document units like sections and paragraphs. The macros that define the beginnings of sections take a heading argument that is the name of the section. In the Cognitive Science Laboratory at UCSD we use section macros based on the American Psychological Association publication guidelines (APA 1975). These have macros for high headings, main headings, left headings, and paragraph headings, each being logically nested in preceding ones. In other documents a numerical argument to the section macro indicates the level or depth of the section. Such a scheme is used in this document. A heading outline for a document is a graphical abstraction in which: each section heading occupies one line, headings are indented proportional to their depth, and all other text is removed. Numeric indices might be included, as might other information about the section headings. Optionally, paragraph beginnings can be indicated. For example, the heading outline for a draft of this document is:

1 Good Technical Writing
2 Traditional Aids: Summary/Predictive Statistics
3 A More General Approach: Abstractions
4 Two Graphical Abstraction Programs
    4.1 HEADINGS: Extract Section Headings
    4.2 PUNC: Punctuation Graphs of Sentences
    4.3 ABSTRACT: Combining the Two Programs
5 Conclusions
6 Acknowledgements
7 References

The headings outline allows a writer to see the overall organization of a document. By skimming the outline vertically, the number of sections at any level is apparent. It is possible to observe the variation in section length by the appro-

priate selection of options. Writing techniques like parallel development, where the same topics are expanded under each section, can be verified.

4.2 PUNC: Punctuation Graphs of Sentences

A punctuation graph of a sentence is a graphical abstraction in which: sentences are displayed one per line, each word is replaced by an underscore, and punctuation is maintained verbatim. Optionally, certain classes of words can be highlighted with something other than the underscore. For example, capitalized words, pronouns, prepositions, etc. can be represented by other characters, or word length can be represented. For example, the punctuation graph of the first sentence of this section is show below.

_____:_____,_____,_____.

Punctuation and sentence length are retained, and everything else is discarded.

The punctuation graph for a sentence shows sentence length and complexity in a way that is easy to grasp. Long sentences literally stand out from the rest, and complex sentences, often heavily punctuated, stand out because they look "busy" compared to the rest. Parenthetical remarks (like in this sentence), lists, "quotes," and the like, are easy to distinguish; see the punctuation graph for this sentence.

__(_____),_,"_,"____,_____;_____.

Decisions about sentence acceptability can be made quickly and based on more information than a readability score. A writer might decide a sentence is acceptable because it is a list. By examining the punctuation graphs for a document writers can observe the change of sentence structure over the length of a document. To help writers find problematic sentences the program can be directed to print the document line numbers of sentences longer than some criterion.

4.3 ABSTRACT: Combining the Two Programs

The UNIX system makes it easy to combine programs to do novel tasks. The HEADINGS and PUNC programs can work together to provide a more sophisticated abstraction of a document. The ABSTRACT program combines the two by showing section headings with sentences replaced by their punctuation graphs. ABSTRACT shows all information of its component programs, but also gives better information about section length, and where sentences are located. The combination is simple: both programs can print input file line numbers with their outputs so all ABSTRACT does is call the UNIX sort facility on their combined outputs. An abstraction of a draft of this document is shown in Figure 2.

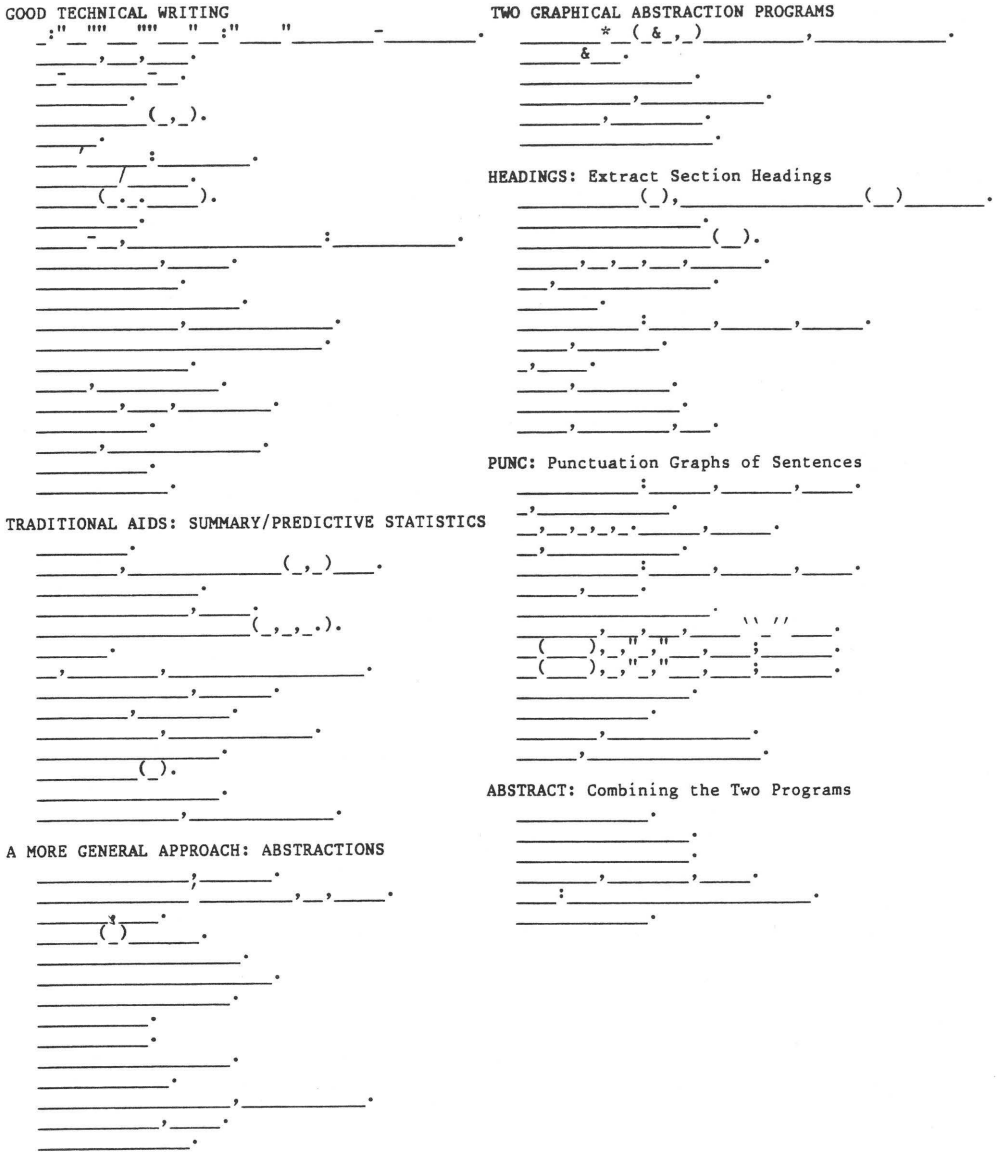

Figure 2. Abstraction of this document. The combined interleaved outputs of the HEADINGS and PUNC programs for a draft of this document. Note how combining the two programs allows the writer to estimate section length.

5. Comparison of Graphical and Numerical Techniques

We think a direct comparison of graphical and numerical representations for text is important for deciding their relative merits. To do this we keep in mind one basic question: What information can one technique represent that the other cannot?

5.1 Graphical Representations of Readability

First we will consider whether graphical techniques can visually represent information integrated into readability scores.

Sentence Length: This is encoded as the length of a punctuation graph because each word is represented by an underscore.

Word Length: This is encoded as a digit representing the word length, but can also be represented by vertical or horizontal bars with lengths proportional to word length.

Sentence Complexity: Compound, complex, and compound-complex sentences can be highlighted in many ways, perhaps the easiest being a single character attached to a sentence graph, or some sort of brightness manipulation possible on most CRT terminals. Difficulty of reading would be represented visually as unusually dim or bright documents.

Sentence Beginnings: Sentence beginnings, as well as words of different classes in any sentence position can be highlighted with color or special characters. For example, expletives could be highlighted in one color while verbs could be in another.

One important point is that the graphical displays are extensible, and that even with simple terminals, they can represent statistics.

5.2 Deficiencies of Numerical Techniques

Now we will point out cases where statistics fail to distinguish sentences and even whole documents that vary greatly in their readability. Numerical statistics do not attend to structural information that help readers visually parse sentences and documents.

5.2.1 Sentential Analysis: Parenthetical Remarks

At the sentence level, the statistics do not discern when parenthetical remarks are used. As an extreme example, the following sentence is likely to be mistaken for a difficult one:

> The need for good nutrition is widely acknowledged (Jones, 1822; Filbert, Able, & Swine, 1924; Feeble & White, 1942).

By parsing the information in the parentheses, readability score based programs can miss by several grade levels. The following PUNC graph of the sentence shows citations in a pattern familiar to PUNC users.

```
_____(_,#;_,_,&_,#;_&_,#).
```

It lets people decide that the parenthetical part of the sentence is acceptable and can be ignored. More common cases are when PUNC graphs show a sentence to be broken up by parenthetical remarks. Two PUNC graphs shown below indicate two sentences of equal length, equal average word length, and so on, that a readability score program will not distinguish.

```
_____#_____&_____.
_____#__(_____)_____(_____)&_____.
```

The first graph shows a sentence about fifty words long (which is commonly thought to be a bad idea) while the second shows a sentence with the same words broken up by parenthetical remarks (which can be ignored somewhat) and this can help a reader with a sentence. The sentence graphs are for the previous sentence. The important point is that both look like they can cause readers problems, but that a writer can decide based on the PUNC graph that the parenthesized sentence is more acceptable, or that the parenthetical remarks should be removed. A readability score does not distinguish between the two, and for good reason; how could the text inside the parentheses be weighted in the readability score?

5.2.2 Sentential Analysis: Lists

Another case where readability grades do not fare well are in processing lists. Lists add to sentence length, substantially when list items are phrases rather than individual words, and this in turn adds to a readability score. This is contrary to research that has found lists easy to read, especially when displayed in a tabular format (Horn, 1983). Lists are discernible in PUNC graphs by the presence of repeated commas or semi-colons, often preceded by a colon. Some examples are

```
_____:_,_,_,_,&_.
_____(_,_,_,_,_)____(_,_,_,_,&_).
__,_____:_____;_____;_____;&___.
```

Again, this is a case where punctuation inserted to help readers is made apparent with a graphical display. Although the last PUNC graph represents a long sentence, the writer might decide it was acceptable because it is obviously a list.

5.2.3 Document Analysis: Headings and Paragraphs

Kirk and Spock search for the crucial information.

Spock: Here it is, the complete knowledge of the Fibrini.

Spock pulls out a huge tome.

Kirk: Is it indexed?

Spock: Yes . . .

Spock finds the information and saves the planet.

As this excerpt from an episode of Star Trek illustrates, the readability of a document can have little to do with readability of individual sentences. It may be more efficient to read small sections of poorly written text than large sections of well written text. Sections and paragraphs greatly add to the skimability of documents. The combination of the HEADINGS and PUNC programs shows the structure and relative sizes of sections in a form convenient for fast verification. Traditional statistics do not discriminate between documents with good or bad or even no structural information. Adding weighted measures of paragraph and section size and structure is an obvious solution, but the measuring and weighting of these factors into readability is not straightforward. We prefer to allow writers to *see* unusually short or long sentences or sections.

6. Conclusions

These programs are useful for document analysis for the main reason that we, as writers, do the evaluative analysis. Abstraction facilitates our analyses by stripping away irrelevant information, allowing us to focus our attention on particular aspects of the text. Large amounts of information can be summarized in simple graphical displays. Abstraction can also help us find specific problem areas: the punctuation of the fourth sentence in the second section; the headings of the third section. Graphical summaries are not sensitive to text length as are statistical ones, and do not depend on a writer's understanding of numerical metrics.

The programs here are not a complete set of text analysis programs. They are a sample of the sorts of analyses we would like to be able to do. More sophisticated graphics might allow better representations of text, but we think new, not only better, graphical displays are needed. More sophisticated programming and integration with text editor programs might allow programs like these to be used actively *during* the writing process, rather than a *post hoc* analysis. Our programs are fast enough to allow interactive use with existing software, a criterion we consider necessary to motivate people to use them. We hope our examples can point the way for more novel abstraction programs, especially those with a graphical flavor.

Acknowledgements

Bob Glushko has stirred up many of the objections we now have for traditional writer's aids. Don Norman wrote the original version of the HEADINGS program. It was re-written to add greater functionality and make it more efficient.

References

The APA Publication Manual (3rd Edition), The American Psychological Association, Washington, D.C., 1975.

Cherry, L. L., & Vesterman, W. Writing Tools—The STYLE and DICTION Programs (Computing Science Technical Report #91). Murray Hill, NJ: Bell Laboratories, 1980.

Coke, E. U. Computer Aids for Writing Text, in *The Technology of Text: Principles for Structuring, Designing, and Displaying Text,* D. H. Jonassen (Ed.). Englewood Cliffs, New Jersey: Educational Technology Publications, 1982.

Horn, R. E. Structured Writing and Text Design, in *The Technology of Text: Principles for Structuring, Designing, and Displaying Text,* D. H. Jonassen (Ed.). Englewood Cliffs, New Jersey: Educational Technology Publications, 1982.

Richie, D. M. and Thompson, K. The UNIX Time-Sharing System. Bell System Technical Journal, 57, 1974, 1905-1929.

Wright, P. Manual Dexterity: A User-Oriented Approach to Creating Computer Documentation. Proceedings of the CHI '83 conference on human factors in computing systems. December, 1983. Published by the Association for Computing Machinery.