Abstraction Program Aids to Documentation

Gary Perlman
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, New Jersey, 07974 USA

Thomas D. Erickson
Department of Psychology, C009
University of California, San Diego
La Jolla, California, 92093 USA

We describe two programs to help writers get an overview of a document so they may judge how well it is organized and how sentences are structured. The HEADINGS program prints section headings in a variety of formats such that indentation shows nestedness. The PUNC program prints punctuation graphs of sentences that show sentence length and complexity. The two programs can be combined so that writers can make estimates of section lengths. Though simple, the programs provide a useful source of feedback telling writers some abstract summaries of their documents. The programs do not make judgements about how the outputs *should* look because they depend on the type of document, however we discuss some of the common trends to look for.

1. Introduction

When we write documentation for programs, we know we are writing for an audience that does not want to spend their time reading about programs. They want to find out what they need to know to use a program and get on to something substantive. The same is often true for other forms of technical writing, but it is always the case with program documentation. This makes good organization and easy indexing to relevant sections one of the most important aspects of documentation. Of course, the text within a section should be easy to read, too. Sentences should not be too long, and there should not be too many sentences inside a section.

In this paper, we describe two programs we use to help us measure document structured so that we may improve it. Before describing the programs, we should describe our document preparation environment. We use the **troff** system (Kernighan, Lesk, & Ossanna, 1978) which is a general purpose text formatting language with macro definition capabilities. Text formatting commands are interspersed in text on separate lines that begin with a period followed by a two character command. For example, the following lines tell **troff** to use an indent of one inch, double line spacing, and a bold font.

```
.in 1i
.ls 2
.ft B
```

Most of our writing is structured with high-level macros defined in terms of primitive formatting functions. We might define

section headings to be preceded by two spaces, centered, and underlined, with a macro definition like the following.

```
.de HD
.sp
.ce
.ul
..
```

At San Diego, we have used a collection of **troff** macros to adhere to the publication standards for the American Psychological Association. These feature nested section heading macros for highest-level-headings (hh), main-headings (mh), left-headings (lh), and paragraph-headings (ph). This document is written with a different set of heading macros whose depth is indicated by a numerical argument to a macro.

2. The Programs

The purpose of the programs is to provide writers with different high-level overviews of the documents they are working on. As this paper is being written, it is difficult to maintain an overview of the structure, even though the structure is quite simple. For larger, more complicated documents, it is not humanly possible to pick out the section headings and understand the nested structure based on the raw **troff** input. Even the formatted copy is inadequate because it may a long time to process, and it may be difficult to judge structural deficiencies from sections that span several pages.

2.1 Section Organization

The HEADINGS program extracts any style of **troff** section heading macros and creates an outline in which headings of sub-sections are indented under headings of super-sections. Program options allow section numbering, identifying paragraphs, and locating the sections in the source files. Because the text is not formatted, HEADINGS can process several thousand lines of text per second, so HEADINGS can be used as an on-line writing aid.

Even for a simple document like this one, the HEADINGS output indicates good structuring.

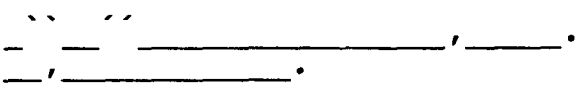
```
Introduction
The Programs
  HEADINGS: Section Organization
  PUNC: Sentence Punctuation
  ABSTRACT: Document Summarization
Conclusion
References
```

We can see that no section is much more detailed than any other, and that no section has too many sub-sections.

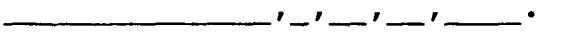
2.2 Sentence Punctuation

A ``punctuation graph'' of a sentence shows a sentence on one line such that words are replaced by underscores, and punctuation

is maintained verbatim. For example, the punctuation graphs for this and the previous sentences are shown below.



Punctuation graphs give us information about sentence length because long sentences stick out from the rest. They also tell us about sentence complexity by showing the punctuation that delimits clauses, quotes, parenthetical remarks, or lists, as in this sentence.



Long, unstructured sentences can show up as a single long line with no punctuation. Sentences with unusual complexity stand out by appearing ``choppy.''

The PUNC program extracts sentences from a **troff** source file and transforms the words. Optionally, it can identify certain classes of words (connectives, negatives, numbers, etc.) and where in the source files the sentence ends. Because there is no semantic and little syntactic processing, the PUNC program can process several thousand sentences per second, so line HEADINGS, it can be used as an on-line aid to writing.

2.3 Document Summarization

The HEADINGS and PUNC programs can be combined to provide more high-level information about a document. Both the programs maintain input file line numbers so their outputs can be combined and sorted so that section headings are correctly located in the punctuation graphs. The result is a high-level overview of a document that tells us section lengths.

The output of the ABSTRACT program at the end of this paper shows the trend of section sizes though different sections. Most of the detail is dedicated to the programs, and there tends to be more introductory material than summary or transition. The section lengths for the individual program descriptions is an accurate summary of my intentions. From the ABSTRACT output, we can conclude that at least one aspect of this document is satisfactory.

3. Conclusion

Though our implementations depends on the **troff** formatting language, they do not depend on any particular macro package. The programs are simple enough that they could be rewritten for other text processing systems like TEX (Knuth, 1979) in any convenient programming language. We think it is important that the programs are fast enough that writers will use them without concern about how long it will take. We also think it is important that the

programs present their summaries uncritically and allow the writers to make their own decisions. Though this last property might imply the programs are only useful to skilled writers, we think that with limited practice, they can be useful tools for improving the structure of technical documents.

4. References

1. Kernighan, B. W., Lesk, M. E., & Ossanna, Jr. J. F. *Document Preparation.* **The Bell System Technical Journal. 57:6.2**, 1978, 2115-2135.

2. Knuth, D. E. **TEX and Metafont: New Directions in Typesetting.** Digital Press, Bedford, Mass., 1979.