Information Research, Vol. 3 No. 4, April 1998

Human creation of abstracts with selected computer assistance tools

Timothy C. Craven
Faculty of Communications and Open Learning,
The University of Western Ontario,
London, Ontario N6G 1H1
Canada

Abstract

After introductory training, a research assistant used the TEXNET abstracting assistance software to create abstracts to articles available via the World Wide Web. The assistant also compiled introductory documentation, including a guide to abstracting using computer assistance tools. This article discusses problems encountered, tools selected for preferred use, and implications for future software development.

Introduction

Suggestions for purely automatic abstracting methods, as surveyed by Paice (1990) and Endres-Niggemeyer (1994), do not show immediate promise of totally superseding human effort. An appropriate short-term goal would seem to be a hybrid system, in which some tasks are performed by human abstractors and other tasks by software. The model of such hybrid abstracting with which this paper is concerned involves providing writers of conventional abstracts with various computerized tools to assist them.

The general aim of the research reported in part in this paper is the development of a prototype computerized abstractor's assistant. As a kind of writer's assistant, such a software package should encompass a simple word processor and other general writer's tools (Kozma, 1991). In addition, the package should integrate tools, such as an automatic extractor, related specifically to the task of abstracting. Abstracting assistance features are being prototyped in a text network management system, known as TEXNET (Craven, 1988) (Craven, 1991b).

Little other research has been reported on the development of computerized abstractor's assistance packages. By contrast, there exists a wide range of assistance packages for indexers, including such commercial products as CINDEX (Indexing Research, 1997), MACREX (Calvert, 1998), and SKY Index (SKY Software, 1998). Some specialized applications, such as that developed for the American Petroleum Institute (Martinez, Lucey, and Linder, 1987), analyze free text and suggest suitable controlled-vocabulary index terms.

Among other purposes, TEXNET is designed to allow abstracting of Web pages. It does not, however, include a Web browser. Rather than reinventing the wheel, it has been decided to modify TEXNET to work as a companion program to Netscape Navigator or other compatible Web-browsing software. A similar approach is exemplified by WEBNET, a companion program for the the WWW browser, that provides a graphic display interface mapping areas of the Web visited (Cockburn & Jones, 1996).

When properly configured, TEXNET can capture HTML pages from a compatible viewer and translate them into its own format, automatically assigning weights to text segments in the process. The user invokes the capture and

translation by clicking on a button that is controlled by TEXNET but that floats on top of other application windows, including the browser.

Should compatibility problems arise with the browser, another option is available: TEXNET allows translation of any HTML code stored on the Windows clipboard, via a "Filter Clipboard" function.

How the various HTML tags are translated and what weights are associated with text tagged in various ways are controlled by a set of filters. These filters have default values and can also be edited by the user.

One method of testing the value of the TEXNET tools, or any tools for assisting in the writing of abstracts, is formal experimentation. In an ongoing series of experiments (Craven, 1996), subjects have been asked to write abstracts of an article using a modified version of TEXNET from which many features have been omitted. Features that are included, and on which the experiments have focused, are automatically generated displays of keywords or phrases.

Of these two displays, that of keywords was the first to be developed, for two main reasons. First, an earlier study (Craven, 1991a) had showed little use in abstracts of longer verbatim word sequences from full texts. Second, keyword extraction is a somewhat simpler task than phrase extraction, though methods for efficient phrase extraction do exist, as in INDEX (Jones *et al.*, 1990), FASIT (Burgin & Dillon, 1992), CLARIT (Paijmans, 1993), and work reported by Fagan (1989). The key phrase display was developed later (Craven, in press). A special feature of this display has been the use of a compact form that takes account of overlaps among the phrases.

Results of the experiment have showed considerable variation among subjects, but slightly fewer than half of those presented with the phrase display have found it "quite" or "very" useful in writing their abstracts.

The experimental situation has had clear limitations. Subjects have had a very limited amount of time to become familiar with the features of the abstracting assistance tools. To allow any familiarity within the time available, they have been given access to only a very limited set of tools. Their previous experience with abstracting may have been scanty or nonexistent. The source texts, while selected so as not to be excessively esoteric, may have been in unfamiliar areas.

Accordingly, a different, more extended kind of evaluation and testing of the current state of TEXNET was called for, which, it was hoped, would serve in part to complement the experimental results. The emphasis would be on the package and its features, rather than on the human users.

Tasks of the research assistant

A student on a co-operative work/study programme was hired as a full-time research assistant for a period of four months. The research assistant had a number of tasks, to be undertaken in consultation with the researcher:

1. to become familiar with the main features of TEXNET;
2. using TEXNET to practice writing abstracts of documents available in HTML format on the World Wide Web;
3. to identify bugs or other problems with using the software in practice;
4. to note which tools or combinations of tools appeared most useful in abstract writing; and
5. to prepare documentation for TEXNET.

The assistant, who had received a little previous instruction in abstracting and was broadly familiar with the guidelines of the then ANSI standard on abstracting (ANSI, 1979) (since superseded by a new edition: NISO, 1997), began work by reading about 25 papers written by the researcher and others on computerized abstracting and on tools for abstractors, and then, as the work period progressed, consulted Cremmins' standard work on abstracting (Cremmins, 1996). In general, the abstracts produced were not evaluated at the time by the researcher; but feedback on the general form considered suitable was provided at about the 6-week point. No exact maximum abstract length was required.

Choice of the specific material abstracted was left up to the assistant. Nevertheless, it was required to be papers or articles, as opposed to other formats that might be encountered on the Web, such as tables of contents, excerpts, or abstracts; it was also not to be too specialized to be readily understood.

A commercial spell checker (Winspell: R&TH, 1994) was made available as a companion program to TEXNET for the assistant's use. This package is capable of monitoring text as it is typed into a window and signalling the user if a word typed appears to be misspelled. Several participants in the experiments had commented on the desirability of spell checking as an assistance feature.

Over the years of its development, TEXNET has acquired a wide variety of features, some easier to learn and use than others. It was decided that the assistant should not explore in detail, or receive much training in, a number of the less likely useful or more difficult features of the package. Generally, these same features were also omitted from the documentation produced, or not dealt with in much detail.

Features to which the assistant paid especial attention included the following:

- importation of Web documents
- displaying and editing of source texts and abstracts
- extracting of text segments based on Boolean queries, on any matches to keywords from a given list, or on segment weights
- weighting of text segments by occurrences of frequent keywords or of keywords from selected passages
- frequent keyword and key phrase displays

Features not emphasized included the following:

- manual or automatic structuring of source texts with links that reflect dependency of one passage on another, contextual passage for its meaning
- database functions, which were designed to work with structured texts
- features requiring a lot of time to develop ancillary data to support their effective use, including a thesaurus and lists of positive and negative cue words and of general indicative formulas or phrases commonly used in abstracting.

Results

As a first set of texts to be abstracted,the assistant chose 30 documents from Seeker1's Cyber Anthology Home Page (then available at http://www.cla.ufl.edu/anthro/cyberanthro/newhome.html - ***[no longer available, December 2002]***). These covered various aspects of human-computer relationships, from cyberpsychology to technoshamanism. Later, abstracting work was performed on 10 documents from Sarah Zupko's Cultural Studies Center Web site (then available at http://www.msc.net/~zupko/articles/national.html, now (***December 2002***) at http://www.popcultures.com/.

Actual draft abstracts were produced for twenty-four of the documents in the first set. Interesting from the point of view of future abstracting tool design are the other elements that the assistant decided to store in what would normally be files for the abstracts alone:

- notes (about ten files),
- longer extracts (about nineteen),
- keyword lists (about fifteen),
- phrase lists (about fourteen),
- definitions (about four),
- and segment weights.

A few technical difficulties were naturally encountered. Some of these led to minor corrections of the software, either during the period of the assistant's employment or later. Others related to specific peculiarities of the hardware or the operating environment used. Those interested may obtain further details from the author.

Initially, the assistant found the displays of frequent keywords not to be particularly useful. The phrase displays seemed of much more value, giving such terms as "think tanks", "pr firms", and "right wing". Somewhat later, however, the small number of substantial phrases in the phrase displays was remarked upon.

The phrase display format was not found to be perfectly self-explanatory: the role of the semicolon (";") in marking phrase boundaries needed to be explained by the researcher, as did the fact that a phrase might consist of only one

word. The original seven word limit on phrase length was later increased to ten, but it might have been desirable to be able to accommodate still longer sequences such as the 12-word "and meant to be played by two refined men in a civilized".

Another respect in which the phrase display was obscure was in the rules followed in deriving it. These had not been explained in the existing help files for the package. In addition, the assistant had already had some exposure to software that extracted phrases in a different manner, based on phrase frequency alone rather than a combination of phrase frequency with word significance.

Certain properties of the source text could degrade the phrase display. For example, a number of repetitions of identical citations in one article led to the phrase list's being almost swamped with sequences such as "Baudrillard 1975".

Both the phrase display and the frequent keyword display were often cut off at the 100-node threshold (with one node for each word and each phrase boundary). This problem could largely be avoided by the assistant's increasing the default minimum number of occurrences for a word to be considered "frequent" in the text. The package was subsequently modified to provide more compact displays with a much higher limit.

At the seven-week mark, the assistant proposed the following as the ideal preliminaries to writing an abstract: read the article; "structure" the text (simply in order to make the text segments generally shorter); call up the frequent keywords display; record selected keyword pairs that seem to represent important concepts; return to the unstructured text (which has normally longer text segments); use the Boolean extract function to view an extract based on each of the keyword pairs previously selected. Some specific personal names, titles of books, and the like might still need to be picked up from elsewhere in the source text. It was also acknowledged that longer, more complicated texts might require a different approach.

By contrast with the frequent keyword display, the automatic manipulation of segment length by switching between structured and unstructured formats did not appear to have much influence on the phrase display.

Manual manipulation of segment breaks was also found useful in improving the quality of extraction.

The file format for the documentation had not been decided on definitely in advance. After the assistant's first draft was ready, it was clear that the material would be best presented as a hypertext and that illustrations would need to be incorporated. The format decided upon was HTML, on two main grounds: creation of Windows Help files with available tools had proven immensely difficult in earlier experimentation; and it was expected that almost all users would have access to an HTML viewer. Use of HTML would also enable the documentation to be viewed by interested parties on platforms that were not Windows-based, even if they could not actually run the software.

Discussion

In dealing with materials on the World Wide Web, it has been found that inverse document frequency scores of terms are not stable (Srinivasan, Ruiz, Lam, 1996). This finding serves as a reminder that stop lists, such as that used in TEXNET, cannot be universal and may indeed need to be modified over time even for the same type of material. As it was, the assistant was given the default TEXNET stop list and was not asked to develop it further. Thus, the feature of TEXNET that allows for stop list variation was not tested in this research.

Development of a set of phrases commonly used in abstracts was also not one of the tasks required of the assistant. Like a stop list, such a phrase list should probably vary with the field, as suggested by Tibbo's (1992) observation of considerable variation in abstract content types between disciplines.

Compared to social science literature, texts in the hard sciences tend to have more domain-specific terms, and more of these tend to occur in sequences (Haas, 1997). This difference might contribute to readier automatic extraction of useful specific indexing terms from hard science texts. Word sequences that might be useful in abstracting, however, as noted elsewhere (Craven, in press), are not limited to phrases resembling indexing terms, still less to terms that are specific to particular disciplines. Word sequences automatically extracted from the documents examined by the assistant included "information wants to be free" and "if you want to see the future"; scarcely suitable as index terms, these clauses nevertheless tell a great deal about the content of the articles from which they were extracted and might even be included in the abstracts.

The switching between structured and unstructured formats suggested by the assistant was later rendered unnecessary by a modification of the word frequency computation method: multiple occurrences of a word within a segment are now counted separately.

In response to comments made by earlier research subjects, an optional dynamic word-count display had been introduced into the full package, for both abstract and full text. The assistant did not mention this feature in any of the documentation prepared. Two factors may be at work here in the way that the assistant was working in contrast to the experimental situation: the source texts had already been filtered to some extent for length by the assistant, and no strong time or space constraints had been imposed.

For a variety of reasons, it does not seem desirable to lock the abstractor out of manipulations of the source text, though a copy of the original should normally be retained in a backup file. Examples noted where temporary modification of the source might be useful include the following: adding or deletion of segment breaks to improve the quality of the automatic extracts; tidying up of anomalies arising in translation from HTML; and deletion of material, such as citations, that may detract from the quality of frequent keyword and phrase displays, as well as from some of the weighting functions.

Further research

The default minimum number of occurrences for a word to be considered "frequent" was initially determined for shorter documents than those used in this study. Automatic resetting of this value when a source text is loaded has now been added to the package. The reset value is currently based simply on the number of lines in the document. There are other variables that might be taken into account in the calculation: the number of words, though this would tend to be highly correlated with the number of lines; the proportion of stop words, since occurrences of these would not be counted; and the repetition of vocabulary in the text, since high repetitiveness could result in too many words being "frequent". As has long been recognized, different texts show different degrees of concentration, or repetition, of vocabulary. For example, vocabulary has been observed to be more concentrated in theoretical than in practice-oriented material (Losee, 1996).

Making available a database of source texts and abstracts seems likely to prove beneficial from several points of view. Directly, abstractors can, for instance, check previous work by themselves or other abstractors or identify more readily the distinguishing features of the document at hand. The software could make use of keyword occurrence counts, dynamically updated, to pinpoint more accurately which words, and hence which phrases, are more likely significant in a given document. Given a sufficient collection of well written abstracts, it might even be possible to begin to suggest suitable phrasing for the abstract of a newly added document based on past correlations.

As the number and length of phrases extracted increases, especially with longer source texts, so does the need to organize them in a way more useful than mere alphabetical order. Some of the participants in the experiments have already suggested an ordering that reflects the order of ideas in the original or that suggested for the abstract itself. Some preliminary research is under way to investigate the feasibility of achieving this end by simple automatic means.

The assistant's decision to commandeer the abstract files to hold a variety of other useful data about the texts abstracted suggests a strong need that should be met more formally. It should be made easy for the abstractor to store data such as notes, lists of keywords and phrases, extracts, and definitions for an abstracted text. These elements might be assigned either to separate files or to fields within a single file or abstracting record.

As noted by Nielsen (1994), a relatively small number of testers is sufficient for identifying the majority of defects in a software package. It is almost certain, however, that defects do remain in the TEXNET package.

The preferences of one student abstractor cannot necessarily be generalized to all other abstractors. Indeed, in studies involving think-aloud protocols (Endres-Niggemeyer, Waumans, & Yamashita, 1991), it has been noted that individuals use quite different approaches in writing abstracts. This variability of approaches has also been noted in the TEXNET experimental sessions. One question that might thus be addressed is the extent to which other abstract writers find the suggestions in the documentation useful in their abstracting.

To this end, and to provide opportunities for independent evaluation of the TEXNET tools, both the software and

the documentation are being made freely available to students and other interested individuals.

Availability

The TEXNET software referred to in this paper, as well as the simplified version used in the experiments, is written as a Microsoft Windows application in Borland Pascal with Objects 7.0. Either source or executable code is available by sending a 3 1/2" dual-density diskette to the author: both may be obtained if two dual-density or one high-density diskette is sent. An executable version of TEXNET may also be downloaded in a ZIP file from the Web site netlib.slis.uwo.ca.

Acknowledgement

Research reported in this paper was supported in part by individual operating grant A9228 of the Natural Sciences and Engineering Research Council of Canada.

References

American National Standards Institute (1979) *American National Standard for Writing Abstracts* (ANSI Z39.14-1979).

Burgin, R. and Dillon, M. (1992) "Improving disambiguation in FASIT." *Journal of the American Society for Information Science*, **43** (2), 101-114.

Calvert, D. and Calvert, H. (1998) "MACREX HOME PAGE." (http://www.macrex.cix.co.uk/, visited 1998 March 5)

Cockburn, A. and Jones, S. (1996) "Which way now? Analyzing and easing inadequacies in WWW navigation." *International Journal of Human-Computer Studies*, **45** (1), 105-129.

Craven, T.C. (1988) "Text network display editing with special reference to the production of customized abstracts," *Canadian Journal of Information Science*, **13** (1/2), p.59-68.

Craven, T.C. (1991a) "Use of words and phrases from full text in abstracts", *Journal of Information Science*, **16**, 351-358.

Craven, T.C. (1991b) "Algorithms for graphic display of sentence dependency structures", *Information Processing and Management*, **27** (6), 603-613.

Craven, T.C. (1993) "A computer-aided abstracting tool kit", *Canadian Journal of Information Science*, **18** (2), 19-31.

Craven, T.C. (1996) "An experiment in the use of tools for computer-assisted abstracting", *in: ASIS '96: proceedings of the 59th ASIS Annual Meeting 1996, volume 33), Baltimore, Maryland, October 21-24, 1996*, edited by S. Hardin. Medford, New Jersey: Information Today. pp.203-208.

Craven, T.C. (in press) "Presentation of repeated phrases in a computer-assisted abstracting tool kit." *Information Processing and Management*.

Cremmins, E.T. (1996) *Art of Abstracting*, 2nd Edition. Arlington, Va.: Information Resources Press.

Endres-Niggemeyer, B. (1994) "Summarizing text for intelligent communication: results of the Dagstuhl Seminar." *Knowledge Organization*, **21** (4), 213-223.

Endres-Niggemeyer, B.; Waumans, W.; Yamashita, H. (1991) "Modelling summary writing by introspection: a small-scale demonstrative study." *Text*, **11** (4), 523-552.

Fagan, J.L. (1989) "The effectiveness of a nonsyntactic approach to automatic phrase indexing for document

retrieval." *Journal of the American Society for Information Science*, **40** (2), 115-132.

Haas, S.W. (1997) "Disciplinary varieties in automatic sublanguage term identification." *Journal of the American Society for Information Science*, **48** (1), 67-79.

Indexing Research (1997) "Indexing Research: CINDEX". (http://www.indexres.com/cindex.html, visited 1998 March 5)

Jones, L.P.; Gassie, E.W.; Radhakrishnan, S. (1990) "INDEX: the statistical basis for an automatic conceptual phrase-indexing system." *Journal of the American Society for Information Science*, **41** (2), 87-97.

Kozma, R.B. (1991) "The impact of computer-based tools and embedded prompts on writing processes and products of novice and advanced college writers." *Cognition and Instruction*, **8** (1), 1-27.

Losee, R.M. (1996) "Text windows and phrases differing by discipline, location in document, and syntactic structure", *Information Processing and Management*, **32** (6), 747-767.

Martinez, C., Lucey, J. and Linder, E. (1987) "An expert system for machine-aided indexing", *Journal of Chemical Information and Computer Sciences*, **27** (4), 158-162.

National Information Standards Organization (1997) *Guidelines for Abstracts* (ANSI Z39.14-1997)

Nielsen, J. (1994) "Estimating the number of subjects needed for a thinking aloud test." *International Journal of Human-Computer Studies*, **41**, 385-397.

Paice, C. (1990) "Constructing literature abstracts by computer: techniques and prospects." *Information Processing and Management*, **26** (1), 171-186.

Paijmans, H. (1993) "Comparing the document representations of two IR systems: CLARIT and TOPIC." *Journal of the American Society for Information Science*, **44** (7), 383-392.

R & TH Inc (1994) *WinSpell Version 3.08: the Windows Spelling Supervisor*. Richardson, Texas: R & TH Inc.

SKY Software (1998) "Product Information." (http://www.sky-software.com/prodinfo.htm, visited 1998 March 5)

Srinivasan, P., Ruiz, M.E. and Lam, W. (1996) "An investigation of indexing on the WWW", *in: ASIS '96: proceedings of the 59th ASIS Annual Meeting (1996, volume 33), Baltimore, Maryland, October 21-24, 1996*, edited by S. Hardin. Medford, New Jersey: Information Today. pp.79-83.

Tibbo, H.R. (1992) "Abstracting across the disciplines: a content analysis of abstracts from the natural sciences, the social sciences, and the humanities with implications for abstracting standards and online information retrieval." *Library and Information Science Research*, **14** (1), 31-56.

Appendix: an example of an abstract written using the package

(The assistant's abstracts generally did not reach the stage of final editing. The following abstract, however, has been included in the documentation as an example of the sort of result that can be produced. The original is "The Hidden Centre of the 'Gutenberg Galaxy'" by Steve Mizrach [formerly available at http://www.clas.ufl.edu/anthro/cyberanthro/Gutenberg_Galaxy.html ***but no longer available in December 2002***].)

> This is an attempt to refute the technological determinists' assumption that technology is born by serendipity and does not cause social changes until society decides to make some use of it. Two revolutionary inventions - the printing press and electronic media - were planned by people determined to achieve certain political and social goals. In the book *The Lost Language of Symbolism* Harold Bayley analyzed the heretical content of the watermarks or emblems used by paper makers and then by printing guilds and concluded that probably the printers' guilds knew that introducing widespread printing, just as with the introduction of writing, would create losers and winners, but they might have known who the losers would be beforehand, and possibly planned things this way. The early pioneers

in personal computing had the same agenda: they were fighting against the elitist approach to computing. They believed PCs would put the power of computing (by providing means for unlimited access and sharing of information) in the 'little guy's' hands - just as the printing press meant he could get his hands on what the priests and Schoolmen had already been reading in the 15 century.

(The following phrases were extracted automatically from the source text, using the default threshold: "of writing", "people", "the hidden center of the gutenberg galaxy", "the printing press", "word". The following frequent keywords were extracted: "galaxy", "gutenberg", "people", "printing", "word", "writing".)

How to cite this paper:

Craven, Timothy C. (1998) "Human creation of abstracts with selected computer assistance tools." *Information Research*, **3**(4) Available at: http://informationr.net/ir/3-4/paper47.html

---

Contents **Home**
