CONTENT ANALYSIS AS A WORD-PROCESSING OPTION

UNE ANALYSE DE CONTENU COMME OPTION DE TRAITEMENT DE TEXTES

John M. Carroll
Computer Science Department
University of Western Ontario
London, Ontario N6A 3K7

ABSTRACT

A simple content-analysis program incorporated in a word-processing system can display the most significant sentence of a page of text and give a short list of the more important words. This could help authors write titles, summaries, and descriptor lists. The content-analysis program relies on word frequency, precedence, and co-occurrence as indicators of content significance. Tests show it performs at least as well as some trained indexers.

RESUME

Un simple programme d'analyse de contenu ajouté à un système de traitement de textes peut faire ressortir la phrase-clé d'une page d'un texte ainsi qu'une brève liste des mots les plus importants. Ceci aiderait les auteurs à écrire titres, résumés et listes de descripteurs. Le programme d'analyse de contenu compte sur la fréquence de parution des mots, leur priorité et leur co-occurence comme indices du contenu. Des tests démontrent que ce système est au moins aussi efficace que certains indexeurs spécialisés.

INTRODUCTION

Word processing is a computer-based activity involved in preparing documents such as reports and letters. It is concerned both with text editing and formatting the output. When words or phrases are deleted, inserted or replaced, the words of the resulting text are shifted from their previous lines when necessary so that line lengths remain about the same. Word processing systems give the user control over line spacing and the number of lines per page. Other available options include: automatic page numbering, multiple columns on a page, proportional spacing between letters in a word, justification of the right and sometimes left-hand margins, a form-data-entry format displayed on a CRT screen, and correction of spelling errors. Output is produced on a letter-quality printer. Word processing is often carried out using multiple-access minicomputers, microcomputers, or intelligent terminals of remotely-accessed main-frame systems. (5).

OBJECTIVES

We felt that a text-analysis capability (g) would be a beneficial option for a word-processing system. If a text-analyzer could indicate the most content-significant sentence in a block of text and extract a short list of content-significant words, it would help the author or editor to compose headlines, subheadings, abstracts and lists of descriptors.

FUNCTIONAL SCPECIFICATIONS

The requirement that any text-analyzer option would have to be implementable on a microcomputer meant that we would have to work with limited memory and speed - say 16K, and one micrcsecond (internal clock time).

We would have to accept input on any conceivable subject, from a keyboard or disk file, in upper and lower-case characters, and with no format restrictions - except that we chose to work in English. We worked with blocks of 100 to 300 words. This could easily be extended to accommodate the usual 540-word page size of most documents.

DESIGN CONSIDERATIONS

The earliest achievements in document content analysis were derived from H.P. Luhn's work in word counting (7). Despite decades of counterargument, it is still a fact that there is roughly a sixty percent probability that any non-common multiple-occurring word possesses some content significance. And this word counting was easy to implement within the limitations imposed on our text analyzer.

Word frequency counts are more dependable when words are reduced to common stems (eg. computes, compute, computer, computing, etc. COMPU) (2). There are a lot of good stemming programs around (6) but to save computation time we elected to go with right-hand truncation to five characters.

We made no effort to deal with the alleged problem of synonyms. Such an attempt would most likely have required storing a large thesaurus and this would have violated our memory constraint. For what it's worth, we feel that synonymy arises when an author self-consciously attempts to vary his vocabulary as a rhetorical device. However, by then the author has usually used the word in question often enough so that it would be flagged as significant by word counting.

We used a STOP list consisting of 91 right-truncated words to get rid of the common or form words (eg: THE, TO, OF, etc.) (7). We also provided the user with the abilityto input a MUST list of words whose very occurrence would be regarded as content significant. No MUST lists were used in our subsequent testing.

We chose to forego one of the most significant indicators of word content significance, the relative occurrence frequency (or its mirror image, the inverse document frequency) (3, 4). This measure of content significance involves comparing the number of times a word occurs in a document with the number of times it occurs in a corpus. No matter how one defines corpus, one still winds up having to store a look-up table of incorpus occurrence frequencies. The more broadly defined the corpus, the larger the table. The relative-occurrence-frequency approach was deemed to be unacceptable both because of its requirement for memory and because using it might limit our content-analyzer as to subject matter.

Inasmuch as our primary thrust was to select from a block of text the most significant sentence, we implicitly adopted Luhn's co-occurrence criterion of word importance by weighting each sentence according to the number of non-common multiple-occurring words it contained (7).

We also adopted Baxendale's suggestion that words occurring early in a document were likely to be content-significant (1). We did not adopt her corrollary suggestion that words occurring near the end were likely to be important as well because space limitations would preclude taking a count over the whole of many doucments that might be submitted to the analyser - in other words, as often as not we just won't get around to analyzing the end of a document.

We did not pick up on her idea of looking for prepositional phrases or trying to identify norms, verbs or any other parts of speech. These clues to meaning, valuable as they may be in some contexts, at least in English involve too much table look-up, and too much computation such as syntax analysis, to be feasible as part of an add-on to a word-processor.

DESCRIPTION OF THE ALGORITHM

Our algorithms were implemented in several dialects of BASIC (MAXBASIC on the DECsystem-10; MICROSOFT BASIC on the TRS-80). We found BASIC's string-handling capabilities to be invaluable.

We stored 45 common abbreviations to help decide whether an occurrence of the string "charater/period/space" represented the end of a sentence or an abbreviation.

We used four vectors to store data. Each one had 1.12N components where N is the length of text block we have chosen to examine.

Vector 1 is the text vector. It holds the words of text and the sentence delimiters. It is from vector 1 that the most content-significant sentence will be reconstructed and displayed for the user.

Vector 2 is the trunc vector. It holds the first five characters of each word in upper case. We do our word counting over this vector.

Vector 3 is the work vector. (1) For a STOP-LIST word its contents are zero. (2) For all other words, its contents are equal to the number of times the word occurs provided ti is the first (or only) appearance of the word. (3) For all subsequent appearances of a multiple-occuring word, the contents are: minus sign followed by a pointer to the first occurrence. (4) For a sentence delimiter, the contents are: sentence number times 1000, plus a pointer to the first word of the sentence.

Vector 4 is the value vector (1). The value of a STOP-LIST word or one that occurs only once is zero. (2) The value of a multiple-occurring word is

$$\log E\ (N - I + 1)$$

where log e is the natural logarithm, N is the length of text in words, I is the index of the storage vector. Once this value is calculated for the truncated representation of a word type it is stored in the value vector every time that representation appears in the trunc vector (ie. for all subsequent word tokens). (3) For a sentence delimiter, the value is sum of the values of all the words it contains.

This data structure is illustrated in Figure 1.

Our criteria of word importance are, therefore: (1) not a "common" (ie STOP LIST) word, (2) appears more than once on a page, and (3) is first mentioned early in the document, (4) words found on the MUST list are accorded a value of 4 or 2 * log e (N-I+2), whichever is greater.

Our criterion for sentence importance is that it contain the highest concentration of "important" words of any sentence on the page, that is, have the highest value.

OUTPUT FORMAT

Our output format consisted of the most important sentence on the page, followed by the most important words (those with values greater than four) and, in addition, a list of all proper nouns on the page. An example is depicted in Figure 2.

TEST PROCEDURE

Developmental exercises were done with text input from the keyboard. Testing was done with disk and tape resident text. We used 21 documents. Seven consisted of abstracts from a MARC tape cataloguing library and information science abstracts; and 14 were newspaper articles from computer typesetting tapes used by the London Free Press. Of these seven could be characterized as "hard" news articles, and seven as features.

Our evaluation criterion was as follows: when compared with the average performance of a population of trained indexer abstracters, does the performance of this content analyzer differ significantly from the work done by a small randomly selected subset?

Our population consisted of class of 44 MLS students, in LS-569. Design and Evaluation of Indexing Systems. The experiment was performed near the end of the term.

The students were given the 21 articles and asked to work independently, read each article, and mark the most significant sentence of each article from the standpoint of content.

For each document, a composite ranking of sentences was prepared by scoring one for each first-place mention and rankingthe sentences in decreasing order of their scores. Sentences not mentioned by any of the indexers were ranked as ties for last place. These rank orderings of all the sentences of each document became our control variable.

For each document we ranked the sentences according to decreasing order of the computer-assigned values. These rank orderings of all the sentences of each document became our test variable.

Our measure of the goodness of sentence ranking was the Kendall rank-order correlation coefficient with correction for ties (10). We correlated the control variable with the standard of comparison; and then correlated the test variable with the standard. This procedure yielded 21 pairs of values in the range -1 to +1.

A chi-squared test indicated that we could not reject the hypothesis that the pared differences were normally distributed. The mean difference was .18 and the estimate of the standard deviation was .25 with 20 degrees of freedom (8.2).

We used the paired-comparison Student t-test to determine whether we could reject the null hypothesis that no highly significant (99%) statistical difference existed between the test variable and the control variable (8.1).

RESULTS

The test results indicated that we should reject the null hypothesis of no difference. The mean Kendall rank-order correlation coefficient of the test variable was .76 (S x = .18); the mean of the control variable was .58 (S y = .25). The t test result was 3.30(test) vs 2.53 (criterion .01 at 20 df).

CONCLUSION

Our test results suggest that a simple computer algorithm can extract the most content-significant sentence from a short document as well if not better than a trained indexer can. We believe, therefore, that this rudimentary content analyzer would be a useful option for inclusion in a word-processing system.

FUTURE WORK

Our present research effort is directed towards two goals: (1) enhancing the usefulness of this text-analysis program, and (2) extending our capability for automatically "understanding" the meaning of machine-readable text.

In the first instance we are going to implement this program on a popular microcomputer in such a way as to accept input from the disk-resident files produced during word processing.

Our second goal has less immediate objectives. It is concerned with determining the "aboutness" of a document by abstraction rather than extraction.

Our approach will be to employ a non-deterministic formal grammar whose terminal symbols will be words and marks of punctuation that we find to be recognizable by computer algorithms without unacceptable ambiguity.

We suggest that these symbols may include.

dollar amounts	(A)
abbreviations	(B)
code-groups	(C)
explanations	(E)
figures	(F)
hyphenated words	(H)
high-frequency words	(HG)
long words	(L)
"must"-list words	(M)
proper nouns	(P)
pronouns	(PN)
prepositions	(PR)
pauses	(R)
"stop"-list words	(S)
verbs	(VB)

also:

end-of-word	(W)
end-of-sentence	(ES)
end-of-line	(EL)
end-of-file	(Eof)

For example, a production in this language might assert that an abbreviation, followed by a proper noun, followed by an abbreviation, followed by a proper noun is most likely to be a

personal name (like Mr. John J. Jones); and that a proper noun, followed by a proper noun, followed by a "stop"-list word, followed by a proper noun, followed by an abbreviation is most likely to be a corporate name (Winken, Blinken and Nod Ltd.).

Gaining ability to make reasonable inferences regarding the appearances of grammatical productions in text will be a step towards determining the "aboutness" of a document.

We anticipate that a given sequence of productions will fit into one of a set of contextual formats; each of these formats will enable us to profile documents of a given kind.

Take the case of a new-product announcement. The subject would most likely be the product in question. A company name would probably specify the manufacturer of the product; the presence of a role indicator such as by would make it even easier to decide which sequence of strings denotes the manufacturer.

The role indicator for might preface a list of applications. String sequences incorporating figures and code-groups most probably would be performance specifications for the product.

The "aboutness" of a new-product announcement would thereupon be encapsulated in a stylized profile highlighting items of information like: name, model, manufacturer, specifications, cost, and so forth. These profiles would permit easy retrieval of information needed for comparative analysis of products by prospective purchasers.

The reason we identify pronouns is because we plan to use this information to resolve ambiguities arising from anaphora; we identify prepositions and verbs because we believe they will help us to arrive at reasonable role indications for some of the other symbols.

Our formal language will likely have interesting properties because in addition to being concerned with the appearance of terminal symbols, it will also take account of the "distance" between them, as represented by the number of untagged character strings.

It is not intended that our language will be able to handle all kinds of text. Indeed, there is reason to believe that although a universal text analyzer may be computationally feasible, it will not be economically feasible within the forseeable future.

Instead, we believe it will be necessary to write or at least modify, a grammar each time the need arises to process large quantities of text sharing a similar format. One application we have already discussed is classifying new product announcements.

Further on, it may prove possible to develop a software system capable of generating a specialized non-deterministic formal language form (1) a random sample of the text to be processed, and (2) observations of the behavior of representative human classifiers in handling it.

ACKNOWLEDGMENTS

The author thanks Dr. T.C. Craven of the School of Library and Information Science for permitting his class to participate in evaluating this procedure; Dr. J.M. Tague, also of SLIS, for many stimulating and beneficial suggestions and criticisms; Lindsay McDermid for writing the programs; and F.Y. Tam for helping to analyze the experimental data. This work was supported by the National Engineering and Science Research Council under grant #A7132.

REFERENCES

1. Baxendale, P., Machine-Made Index for Scientific Literature, IBM Journ. of R & D, p. 354, October 1958.

2. Carroll, J.M. and J.G. DeBruyn, On the Importance of Root-Stem Truncation in Word-Frequency Analysis, Journal of the ASIS, p. 368, September-October 1970.

3. Carroll, J.M. and R. Roeloffs, Computer Selection of Keywords Using Word-Frequency Analysis, American Documentation, p. 227, July, 1969.

4. Edmundson, H.P. and R.E. Wyllys, Automatic Abstracting and Indexing - Survey and Recommendations, CACM, p. 226, May, 1961.

5. Lentz, W.E., Word Processing, Canadian Datasystems, Vol. 12, No. 11, p. 78, 1980.

6. Lovins, J.B., "Development of a Stemming Algorithm", MIT Project Intrex Report LSL-TM-353, Cambridge, MA, 1968.

7. Luhn, H.P., "Potentialities of Auto-Encoding of Scientific Literature", IBM Research Report RC-101, Yorktown Heights, NY, May 15, 1959.

8. Miller, I. and J.E. Freund, "Probability and Statistics for Engineers", Prentice-Hall, Englewood Cliffs, NJ, 1965, 8.1 paired-sample *t* test, p. 169. 8.2 goodness-of-fit, p. 202.

9. Salton, D., Automatic Information Retrieval, Computer, p. 41, Sept. 1980.

10. Siegel, S., "Nonparametric Statistics", McGraw-Hill, New York, 1956. - Kendall rank correlation coefficient *tau*, p. 213-219.

Figure 1: Data structure

INDEX	TEXT	TRUNC	WORK	VALUE
1	Free	FREE	2	4.59512
2	fishing	FISHI	3	4.584967
3	Is	IS	0	0
4	nice	NICE	1	0
5	but	BUT	0	0
6	Ontario	ONTAR	3	4.543295
7	can	CAN	0	0
8	no	NO	0	0
9	longer	LONGE	1	0
10	afford	AFFOR	1	0
11	to	TO	0	0
12	provide	PROVI	1	0
13	it	IT	0	0
14	in	IN	0	0
15	an	AN	0	0
16	era	ERA	1	0
17	of	OF	0	0
18	diminishing	DIMIN	1	0
19	natural	NATUR	1	0
20	resources	RESOU	1	0
21	*period	*SENT	100 1	13.72338
22	After	AFTER	0	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
32	free	FREE	-1	4.59512
33	*PERIOD	*SENT	2022	4.59512

Figure 2: Output format

The Key Sentence is:

Ontario is one of the few jurisdictions in North America that doesn't require its residents to have a fishing licence.

The Key Words are:

Free	4.59512
Fishing	4.584967
Ontario	4.543295

The Proper Names are:

Ontario

North

America

END OF SUMMARY