II. THE SMART AUTOMATIC TEXT PROCESSING AND DOCUMENT RETRIEVAL SYSTEM

Michael Lesk

1. Introduction

SMART is a fully automatic text processing and document retrieval system. All input is in natural language form, and full texts of documents may be used. Among the features available for purposes of content analysis are: thesaurus lookup, hierarchical processing, several phrase searching procedures including a complete syntactic analysis, statistical association methods, title/body differentiation, numeric representation of data, and automatic evaluation of the results of any retrieval run. A full description of the methods and programs employed may be found in Information Storage and Retrieval, Report No. ISR-7, The Computation Laboratory of Harvard University (1964). A shorter but more accessible writeup of the system is given by G. Salton in the Proceedings of the ACM National Conference – 64.

The system described here is the initial version of SMART. Further development will include (a) conversion of part of SMART to time-sharing operation using the MIT Compatible Time-sharing System; (b) expansion of the basic system to remove many of the current size limitations and to include new processing features such as author and citation indexing. Operating instructions for these revised systems will be made available at a later time.

2. Machine Configuration

SMART operates on an IBM 7094 with 32K of core storage. Depending on the processes to be performed, up to twelve tapes on two channels may be required. Data channel trap is not used. Any additional special features (e.g. core protect) should be turned off. A helpful, but nonessential feature, is the printer clock (RPQ 78054).

A. Operating System

The monitor system in use at Harvard is a FORTRAN II, version II system with a nonstandard loader. To run SMART at another installation, one may use a prewritten program tape (see Part 3(B)). The tape loading program is simple and a version for any 7094 system can be prepared easily. Alternatively, card decks compatible with either the old or new (June 1964) version of the IBM FMS loader are available.

B. Magnetic Tape Assignments

SMART uses the normal FORTRAN tape assignments with the addition of the SMART library tape, which is placed on tape drive B5 (see Part 4(B)). If a prewritten program tape is used, it is mounted on B3. Additional tapes are required if the options involving syntactic analysis are in use. SMART uses the Kuno Multiple-path Syntactic Analyzer, as described in Mathematical Linguistics and Automatic Translation, Reports No. NSF-8 and NSF-9, the Computation Laboratory of Harvard University (1963). The condensed grammar tape for the analyzer must be mounted on A5, and tape drives A6 and B6 must be available as scratch tapes.

C. Printed Output

The input tape to SMART must be in standard card-image format, including the normal "look-ahead" coding. The output print tape is normally blocked. A version of SMART that writes unblocked print is also available.

1. FORTRAN II Tape Assignments

A1	System tape. SMART will not address this tape (except to rewind it) until exiting. (See Part 3(B)).
A2	Card input. Standard card image format input tape.
A3	Off-line print tape. (See Part 2(C)).
B4	Off-line punch tape.
A4, B1, B2, B3	System scratch tapes. If a program tape is premounted on B3, no compilations or assemblies may be performed during that computer run. SMART uses A4, B1, and B2 for scratch.

2. Library Tape

B5	Any dictionary lookup or hierarchy expansion requires a library tape on B5; any dictionary update requires a blank tape on B5.

SMART Tape Requirements

TABLE 1

3. Syntax Tapes All tapes listed above plus: A5 Kuno condensed grammar 6C. A6 Scratch. B6 Scratch. Appendix 1 contains a list showing the data written on each tape at each stage of SMART. SMART is being revised so as to eliminate tapes A6 and B6. The maximum number of tape drives in the revised version will be ten, five on channel A and five on channel B.

TABLE 1 (continued)

3. Program Loading

Four basic steps are required to run SMART. These are: loading the programs, preparing the library (see Part 4), specifying the processing methods (Part 5), and providing the data (Part 6). The present part describes how SMART is read into memory.

A. Chain Structure of SMART

The system comprises up to twelve chain links, numbered from 1 to 12. Links 10, 11, and 12 are preprocessors whose function is to update program, library, and data tapes. Normal runs begin with link 1.

The SMART program tape normally contains links 1 through 9. This suffices for all runs other than update runs, and represents the easiest way to run the system. A version of the tape which contains the library update links (11 and 10) is also available. The program/data tape update link (12) must be submitted on cards and requires a reasonably standard FORTRAN II loader. Running SMART from a program tape is described in Part 3(B); the loading of chain links from cards is discussed in Part 3(C).

B. SMART Program/Data Tape

The SMART program/data tape is divided into two sections; the programs themselves and data decks of documents (Part 6). The program section occupies the first two files. File one consists of a standard IBM chain tape containing links 2 through 9 (16 records). File two contains one record, a self-loading version of link 1, for use with the standard program tape. The program tape which contains the update links has four more records in file two; these are chain links 11 and 10 with their label records.

To make a normal (nonupdate) run of SMART, the program tape must be spaced one file and "load tape" simulated on B3. A two-card program called TAPLD is available to do this, alternatively, a dozen-card assembly language program can be supplied. If necessary, the operator can perform the tape load from the console. SMART can thus be run under virtually any 7094 operating system.

When a run is terminated, SMART generally tries to execute a FORTRAN II exit. However, if sense switch 1 is down, SMART first attempts to load the on-line card reader. This permits the normal operating system

to regain control. If the card reader on channel A is empty, SMART tries a FORTRAN II exit.

The appearance of the output is improved if a version of TAPLD is supplied to do the following:

(1) leave the date in 142_8 in the form of six BCD characters MODAYR (i.e. 082764 for August 27, 1964) and,

(2) leave the time at start of job in 75_8 in the form of six BCD digits giving the time in decimal tenths of minutes (i.e. 001027 for 102.7 minutes).

The Harvard FMS handles this automatically.

Running an update run of the library tape requires either a program tape containing the update links in file 2, or submission of these links on cards. If the program tape contains the extra four records in file 2, a special version of TAPLD to execute the update links before proceeding to the main part of SMART becomes available. Otherwise, links 11 and 10 (in that order) should be submitted as a normal FMS chain job, followed by normal TAPLD as chain link 1 with all chain links placed on tape 4 (A4).

C. Card Decks for SMART

Normally it is preferable to run SMART from tape, because the full binary deck for the system consists of over 3,000 cards. The obvious exception to this rule arises when updating the program tape itself; card decks for the entire system must then be provided.

Table 2 shows the order and function of the chain links. When running from cards and no updating of the program tape is carried out, a link that is not required need not be submitted. Note that a SMART deck may begin with either link 12 (a program tape update), link 11 (a library update), or link 1 (normal runs).

Link Number	Logical Tape	Physical Tape	Function	See Part
12	2	B2	Update program/data tape.	4(A) (page II-8)
11	4	A4	Thesaurus file update.	4(B) (page II-10)
10	4	A4	Update remainder of library. Cannot be run unless link 11 is included.	4(B) (page II-10)
1	4	A4	Main link. Required for all runs except runs consisting entirely of updating.	5 (page II-31)
2	3	B3	Lookup link. Needed if any English text is submitted. Requires link 1.	-
3	3	B3	Syntax links. Must be run as a unit; require links 1 and 2. Link 3 is a pre-editor for the Kuno analyzer; links 4, 5, 6 contain the analyzer itself; link 7 is the criterion tree routine.	5(A) (page II-32) 4(B) (page II-41)
4	3	B3		
5	3	B3		
6	3	B3		
7	3	B3		

Chain Links

TABLE 2

Link Number	Logical Tape	Physical Tape	Function	See Part
8	3	B3	Edits the answers to requests. Needs link 1.	5(A) (page II-32) 6(A) (page II-44)
9	3	B3	Automatic evaluation. Needs link 1.	6(C) (page II-48)

TABLE 2 (continued)

4. Updating Program and Library Tapes

SMART can use two prewritten tapes of data – the program and document tape on B3, and the library tape on B5. The condensed grammar tape on A5 is also a prewritten tape; SMART, however, contains no procedures for updating it.

A. Updating the Program/Data Tape on B3

As previously mentioned in part 3(B), the program tape is divided into two sections: the program section, files one and two, and the data section, all remaining files. The data section is optional and of varying length. Chain link 12 updates both sections of this tape.

Updating the Program Section of the SMART Program/Data Tape: Chain link 12 is always the first link if it is submitted at all, and is placed on tape B2 (logical 2). The remaining links are expected to be on B3 and A4. Links 2 through 9 must be on B3 (logical 3) in numerical order; links 1 (and 10, and 11 if desired), must be on A4 (logical 4) in the order 11, 10, 1.

Since file 1 of the program tape is written correctly by the chain loader, link 12 merely spaces over it. Link 12 then copies chain link 1 onto the second file of B3, converting link 1 to a single self-loading record.

Link 12 now reads the first control card from A2. If it is a *LINK card (a card with *LINK punched in columns 1-5, column 6 blank, and the remainder of the card irrelevant), chain links 11 and 10 are also copied to B3 from A4. If the first card on A2 is not a *LINK card, links 11 and 10 are ignored and need not be submitted.

Updating the Data Section of the Program/data Tape: The system input tape (A2) may contain a set of data cards to be written on B3. These data files may be read by SMART during later retrieval runs; they are treated as if they were stored on A2, except that natural language text may not be placed on B3 (Part 6(B)). Use of tape data files can save a large amount of card-to-tape operation.

Three control cards on A2 direct the writing of data on B3. All are in normal SMART control card format: an asterisk in column 1, a four letter code in columns 2 through 5, and a blank in column 6. These control cards are *COPY, *LOAD, and *QUIT. *COPY indicates the beginning of a new data file on B3. SMART writes an end of file mark on B3 and copies onto B3 all cards on A2 up to the next link 12 control card. Note that these cards include control cards to parts of SMART other than link 12. Data decks on B3 are used to supply reference document collections to SMART. These decks

contain previously looked-up text as described in part 6(A) and control cards as described in Part 6(B). Typically different files contain reference collections processed by different methods. Note that the last card of any data file should be a *BACK card.

*QUIT terminates the update; SMART writes an end of file on B3, rewinds B3, and exits. *LOAD also terminates the updating of the program tape, but in this case SMART proceeds with the computer run by transferring to the next chain link. SMART will go to link 11 if that link is submitted, or to link 1 if link 11 is not submitted. Note that if link 11 is submitted in a run, SMART proceeds to execute it after *LOAD even if *LINK is not used.

Note also that if link 12 is used, the first card after the FMS *DATA card must be one of the four *LINK, *COPY, *LOAD, *QUIT. If it is *LINK, one of the other three must follow the *LINK card.

B. Updating the Library Tape on B5

The library tape is updated by links 11 and 10. When these links are executed, B5 must contain a blank tape on which the library is to be written. Provision is made for using an older library in the update; files may be copied intact, or minor alterations may be made without resubmitting the entire library on cards. Any previous library to be used must be mounted on tape A6.

The SMART library consists of five files. The data and formats are shown in Table 3. A summary of the library maintenance cards is given here; full details for most files are available in Information Storage and Retrieval, Report No. ISR-7.

File 1:	Record 1.	Label (one word).
	2.	Thesaurus (variable length).
	3.	Label (one word).
	4.	Suffix (variable length).
File 2:	Record 1.	Both records used for suffix list.
	2.	
File 3:	Record 1.	Criterion trees. See Sections 6-8 of Information Storage and Retrieval, Report No. ISR-7.
	2-n.	Further criterion trees.
File 4:	Record 1.	Statistical phrases, up to 100 per record.
	2-n.	Further statistical phrases, blocked 100/record.
File 5:	Record 1.	Label (five words).
	2.	Hierarchy (variable length).

SMART Library

TABLE 3

Link 11 updates the dictionary file; link 10 is responsible for the rest of the tape. If only a dictionary file is being written (i.e. a null thesaurus tape), link 10 may be replaced with a dummy link that calls EXIT, or link 1 (as desired). Alternatively, link 10 may be left untouched and empty files written.

The control cards for updating the library appear on the system input tape, A2. They follow update instructions for link 12 (Part 4(A)), if any, and precede the main instruction deck for link 1 (Parts 5 and 6). Update cards for the five files are submitted in order, and if both links 11 and 10 are being executed, at least one update card for each file must appear in the input deck.

A new library tape is written using control cards from A2, data cards from A2, and/or an old library on A6. Note that some control cards specifically ignore A6, so that this tape may not need to be mounted. If any such control card is submitted, all files must be written without references to A6, since the tape positioning will be incorrect. The purpose of the "ignore A6" control cards is to permit a complete rewrite from cards. A partial rewrite from cards should be done with control cards which do not use the data on A6, but do space the tape forward.

Thesaurus and Thesaurus Suffix List Updating: The first file of the library tape is devoted to the thesaurus lookup. This file contains a stem thesaurus in which each English stem is converted by the lookup into 1 to 6 concept numbers. Any concept number may correspond to any number of English stems. A particularly simple version of the thesaurus is the so-called "null" or "vacuous" thesaurus. This dictionary consists of a one-

to-one mapping from words to concepts, and the only purpose of the lookup is to reduce various forms of the same word to the same stem (and to condense 24 alphabetic characters to 12 binary bits). The "null" thesaurus can be prepared automatically from English texts (Part 8(C)).

The dictionary update programs require one control card. Columns 1-6 of the control card specify which parts of the files are being changed. This field may contain either BOTH, THES, or SUFFIX, corresponding to updating both parts of the lookup file, only the thesaurus, or only the suffix list. If only one part of the file is being changed, columns 7-12 specify either UPDATE or START. UPDATE implies that the cards which follow on the input tape are being added to the file; START, that the current file is being replaced by the input cards. When BOTH is requested in columns 1-6 of the control card, columns 7-12 select UPDATE or START for the thesaurus and columns 13-18 do the same for the suffix list. For example, THES UPDATE adds to the thesaurus and leaves the suffix list unchanged. BOTH START UPDATE rewrites the thesaurus entirely, but only adds to the suffix list. BOTH START START rewrites the entire file from cards and ignores A6. To skip A6 while rewriting, a 1 in column 24 should be added to the BOTH START START card.

If any changes are being made in the thesaurus, the update cards should follow immediately after the control card. They contain an English stem in columns 1-24, with trailing blanks, and up to six semantic concept numbers punched in four column numeric fields from column 25-48 (right-adjusted within each field, but using the leftmost fields first). Concept

numbers less than 1,000 are significant; the numbers above 1,000 are reserved for nonsignificant words. A concept number of 0 is non-significant; concept numbers above 3,072 are forbidden.

The syntactic data associated with a given stem are placed in columns 49-72 in three-column numeric fields (again, right-adjusted within each field but using the leftmost fields first). These numbers correspond to partial homographs for the Kuno Multiple-path Syntactic Analyzer (Part 5(A)). Up to eight homographs may be given for each stem. Table 4 shows the correspondence between partial homographs and syntax codes. The homographs are completed by combining the partial stem homograph with the partial suffix homograph (Part 4(B)) in chain link 3. For greater detail, see Chap. III, Part 3(A) of Information Storage and Retrieval, Report No. ISR-7. The last data card for the thesaurus update contains ZZZZZZ in columns 1-6 to mark the end of the list.

The thesaurus update cards, if any, are followed by the suffix list changes, if any. The suffix is punched in columns 1-12 (in left-to-right order, beginning in column 1 and with trailing blanks). An identifying number is punched in columns 13-15 (right-adjusted). This number is used only to connect this suffix list with the syntactic suffix list described in Part 4(B), and the suffixes may simply be numbered from 1 up.

Neither the thesaurus nor the suffix list need be in any particular order. Since the SMART dictionary lookup is a tree-structure lookup, with the entire dictionary in core, the speed of the lookup is unaffected by

1 ADJ	21 BEOS	41 OI2O	61 HVI	81 QUE
2 ADK	22 BEOY	42 OI3O	62 HVP	82 RL1
3 ADL	23 BGOS	43 OT1O	63 IAD	83 RL2
4 ADM	24 BIO	44 OT2O	64 IAV	84 RL3
5 ADN	25 BPO	45 OT3O	65 IPN	85 RL4
6 ADØ	26 BRO	46 OT4O	66 IPØ	86 RL5
7 ADP	27 CCØ	47 OT5O	67 NAD	87 RL6
8 ART	28 CIF	48 OT6O	68 NØ4C	88 TITS
9 AUXC	29 CMA	49 OT6O1	69 NØ4S	89 TØIS
10 AUXP	30 CØ1	50 OT7O	70 NØUO	90 XCØ
11 AUXS	31 CØ2	51 OT7O1	71 NØVO	91 YCØ
12 AV1	32 CØ3	52 HAVC	72 NUMO	92 NØUS
13 AV2	33 CØ4	53 HAVP	73 PRD	93 VT1P
14 AV3	34 CØ5	54 HAVS	74 PRE	94 VI1P
15 AV4	35 CØ6	55 HP1	75 PRNC	95 NØUP
16 AV5	36 CØ7	56 HP3	76 PRNP	96 NØUC
17 AV6	37 CØ8	57 HP4	77 PRNS	97
18 AV7	38 CPR	58 HP5	78 PRØ	98
19 AV8	39 DØI	59 HPP	79 PRZC	99
20 BEOP	40 OI1O	60 HVGS	80 PRZP	

100 VI1P	110 VI2P	120 VI3P	130 VT1P	140 VT2P
101 II1	111 II2	121 II3	131 IT1	141 IT2
102 VI1S	112 VI2S	122 VI3S	132 VT1S	142 VT2S
103 VI1C	113 VI2C	123 VI3C	133 VT1C	143 VT2C
104 PI1	114 PI2	124 PI3	134 PT1	144 PT2
105 GI1S	115 GI2S	125 GI3S	135 GT1S	145 GT2S
106 RI1	116 RI2	126 RI3	136 RT1	146 RT2
107	117	127	137	147
108	118	128	138	148
109	119	129	139	149

Note: O = "zero," Ø = fifteenth letter of the English alphabet.

Partial Stem Homographs
(See Mathematical Linguistics and Automatic Translation, Report No. NSF-9, Vol. I)

TABLE 4

150 VT3P	170 VT5P	190 VT6P1	200 VT7P	210 VT7P1
151 IT3	171 IT5	191 IT6 1	201 IT7	211 IT7 1
152 VT3S	172 VT5S	192 VT6S1	202 VT7S	212 VT7S1
153 VT3C	173 VT5C	193 VT6C1	203 VT7C	213 VT7C1
154 PT3	174 PT5	194 PT6 1	204 PT7	214 PT7 1
155 GT3S	175 GT5S	195 GT6S1	205 GT7S	215 GT7S1
156 RT3	176 RT5	196 RT6 1	206 RT7	216 RT7 1
157	177	197	207	
158	178	198	208	
159	179	199	209	
160 VT4P	180 VT6P			
161 IT4	181 IT6			
162 VT4S	182 VT6S			
163 VT4C	183 VT6C			
164 PT4	184 PT6			
165 GT4S	185 GT6S			
166 RT4	186 RT6			
167	187			
168	188			
169	189			

TABLE 4 (continued)

the order of either the dictionary or the text. There is one exception to this rule: in any complete rewrite of the suffix list (START option) the first suffix must begin with "e."

A full description of the lookup process is given in Chap. IV of Information Storage and Retrieval, Report No. ISR-7. Note, however, that the lookup is sufficiently accurate to associate (for example) HOPING with the stem HOPE, while HOPPING is found from the stem HOP. Also, if EASE and EASY are stems included in the thesaurus, EASIER is properly connected with EASY, while EASING is found from EASE.

If the first card read by the thesaurus update routine is blank, the first file of A6 is copied to B5 unchanged. No other cards are read by the thesaurus update system in this case.

Syntactic Codes for Suffixes: The second file on the library tape contains the partial homograph codes associated with the suffixes. One control card is required by the library update programs; it contains one of four codes in column 1-5. The possible codes and their meaning are:

(ORIG A deck of suffix cards follows as in Part 4(B); generate a file of these suffixes, ignoring tape A6.

(DUPL The file on A6 is copied unchanged to B5. No other cards are read by this update section.

(MERG Suffix cards following on A2 are merged with the old file on A6.

(SKIP Same function as (ORIG except A6 is spaced over this file.

Suffix Update Cards: These cards contain a suffix number (Part 4(B)) and a list of partial homographs for this suffix. The partial homographs are alphabetic, with the unneeded characters replaced by zeros. For example, the Kuno homograph for a plural noun is NØUP. A noun stem, in our dictionary, is given the syntactic number 070, corresponding to NØUO (O = "zero," Ø = fifteenth letter of English alphabet). The suffix "s" has a partial homograph 000P0. When stem and suffix are combined, the complete homograph NØUP is formed. The stem RECTI, to take another example, has the syntax number 043, which corresponds to 0T10. When combined with the suffix FIED, which has the suffix homograph V00C (among others), a correct homograph VT1C is obtained for RECTIFIED. The remaining homographs are obtained from the other suffix homographs; they are ADJ (obtained completely from the suffix) and PT1 (0T10 + P00 0 = PT1).

To specify the syntax codes for a suffix, the suffix number is punched anywhere in columns 1-6, and the various possible partial (or complete) homographs are punched in six-column alphabetical fields beginning in column 7, and using the leftmost fields first.

Criterion Trees: File three of the library tape is devoted to a phrase dictionary in which phrases are defined by a complete set of structural/syntactic/semantic specifications. This dictionary, called the criterion tree dictionary, is used in a phrase searching procedure which requires that:

(1) the components of the phrase have the proper semantic value, i.e. that they consist of the proper stem, or that the same thesaurus category as the proper stem category be attached to them;

(2) the components of the phrase have the proper syntactic role (e.g. "automatic translation" could be distinguished from "automatically translating"), and

(3) the components of the phrase have the proper syntactic dependency relations to each other (e.g. distinguishing "blind Venetian" from "Venetian blind").

The control cards necessary to prepare a tape with such tree specifications are discussed in the present part. Two distinct formats exist for keypunching the actual trees; before describing these, the overall control cards are described.

Overall Control Cards: The criterion tree editing supervisor recognizes six control cards. These are:

/COPY in columns 1-5, with an optional integer beginning in column 7. The indicated number of trees are copied from A6 to B5. If no number is given, the whole criterion tree file is copied.

/SKIP in columns 1-5, with an optional integer beginning in column 7. The indicated number of trees on A6 are skipped (if no number, the whole file is skipped).

/EDIT in columns 1-5, with an optional integer beginning in column 7. The indicated number of trees are copied from A6 to B5 with the trees specified by the deletion requests (which follow this card) removed from the file. These deletion requests are cards with right parentheses in columns 1 and 2, and with a serial number in columns 7-12, and/or a BCD identifier in columns 13-18. Up to 30 of these delete requests may be given. Any criterion tree on the tape which matches the information on the delete card (either the identifier or the serial number, if only one is specified, or both the identifier and the serial number if both are given on the delete card) is removed from the file.

/WEOF in columns 1-5. This terminates the processing of the criterion trees. The file on B5 is terminated and A6 is spaced over any remaining trees (but see /ADDCD and /ADDL).

/ADDL in columns 1-5. Trees following in the format described in Part 4(B) are written onto B5. If columns 7-12 are nonblank, the normal spacing on A6 produced by the /WEOF card is suppressed.

/ADDCD in columns 1-6. Trees in the format described in Part 4(B) are written onto B5. If columns 7-12 are nonblank, the normal spacing of A6 produced by the /WEOF card is suppressed.

MAKTRE Format (Called by /ADDL): Each criterion tree is represented by one group of consecutive cards. The first card of each group has a six-character BCD name in columns 1-6 and an optional serial number in 76-80 (if given, this serial number must appear on all cards of the phrase).

The remainder of the card group contains at least one card for each node (each node in the tree corresponds to a word in the phrase). The first such node card of each node is blank in columns 1-6, and contains the node number in columns 7-8. These node numbers must begin at 1, and increase by 1 for each node up to a maximum of 36. The root of the tree is marked by an X in column 12. Other nodes have either an I (for indirect dependence) or a D (for direct dependence) in column 12 and the number of the node on which they depend in columns 10-11. These dependency specifications determine the structure of the tree.

Columns 13-72 of the node cards contain the semantic and syntactic specifications. These columns must contain pairs of parentheses enclosing "relation generators." The relation generators are separated by commas; they may be either numbers of more than one digit (including leading zeros) which are interpreted as concept numbers from the thesaurus, or single characters which are taken to be syntactic role codes of the syntactic analyzer. A pair of parentheses enclosing a set of relation generators is called a "relation"; any number of relations may be given each with any number of relation generators. If the information given overflows one card, it may be continued on successive cards with an * in column 12. The data must be placed in columns 13-72 of each card, and the breaks between cards must occur after a right parenthesis.

The interpretation of the tree is made by assuming:

(1) that each node corresponds to one sentence word;

(2) that for each node, the corresponding sentence word satisfies at least one relation generator in each relation (i.e. the relation generators are ored and the relations are anded). Thus, one can specify (S, Ø)(123, 364) meaning "either a subject (S) or an object (Ø), and either concept number 123 or concept number 364," while the relation (S)(123) would mean "both a subject and concept number 123." (See Table 5); and

(3) that the words in the sentence have the same dependencies as the nodes in the tree.

Any sentence which contains a set of words satisfying these conditions for some tree is considered to contain the phrase represented by the tree.

When MAKTRE finds a card with ////// in columns 1-6, control returns to the supervisor of the criterion tree update program.

TRECND Format: An alternative format for trees is available through use of the /ADDCD control card. In this format, the structural and syntactic specifications are taken from a list of thirteen common tree types, and only the semantic data are given in detail. Four pieces of data must be given: the tree identifier, the output concept number of the tree, the relations, and the tree specifications.

Letter	Meaning
1	Declarative sentence
2	Interrogative sentence
3	Imperative sentence
4	Subject clause
5	Object clause
6	Complement clause
7	Adjective clause
8	Adverbial clause
A	Adjective
C	Complement
D	Adverb
E	Adverbial noun phrase
G	Gerund
M	Participle
Ø	Object
P	Phrase
R	Phrase or clause introducer (preposition or conjunction)
S	Subject
V	Verb
X	Auxiliary verb
+	Conjunction (and/or/but)
,	Comma
.	Period
=	Question mark

See Section I of National Science Foundation Report, No. NSF-9, p. 127 (Table 25)

Syntactic Relation Generators and Their Meanings

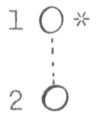
TABLE 5

The tree identifier must be punched in columns 1-6 of the first card for each tree; this identifier may consist of any six characters except six blanks, five blanks and an asterisk, six zeros, or a slash followed by anything.

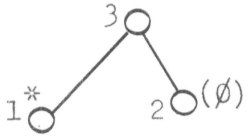
The output concept numbers follow immediately (in contrast to MAKTRE, TRECND does not allow imbedded blanks in its data). The output concept numbers must be integers between 1 and 1,000: there may be up to three output concept numbers, or the whole field may be omitted (note: MAKTRE makes no provision for output concept numbers). Each output concept number is punched as an equals sign, followed by an unsigned decimal integer. This concept number is then attached to the whole tree. Its purpose is to permit finding trees which are made up of trees; the "key node" of the tree (see Table 6) is marked with the output concept number, and will satisfy a relation generator in a later tree containing that concept number.

The output concept numbers, if any, are followed by the relations. Every relation is enclosed within a pair of parentheses as before; however relations corresponding to different nodes are separated by slashes in the TRECND format, rather than being placed on different cards. As in the MAKTRE format, any number of relation generators may be placed within a pair of parentheses separated by commas. Only semantic generators may be used, however. Again, in a valid match, at least one relation generator in each relation must match the sentence node corresponding to the tree node. The relation generators here are as before, unsigned integers separated by commas. A dollar sign terminates the relations. There must be one less slash on the card than there are nodes in the tree; the fields correspond to nodes in the sequence 1, 2, 3 ...

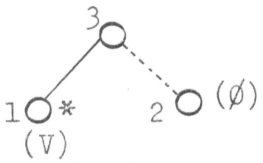
1.



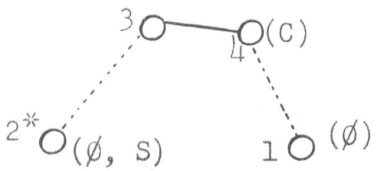
2.



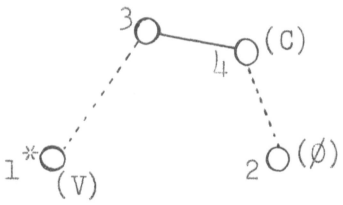
3.



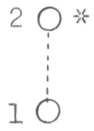
4.



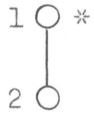
5.



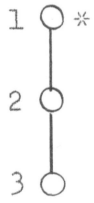
6.



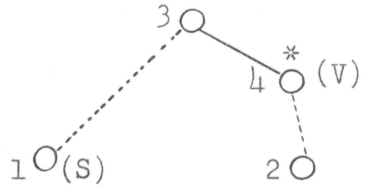
7.



8.



9.



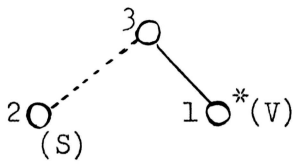
Solid lines indicate direct dependence; dotted lines indicate indirect dependence.

*The asterisk denotes "key" nodes, i.e. the nodes to whose correspondents the output concept number is attached when a match is found.

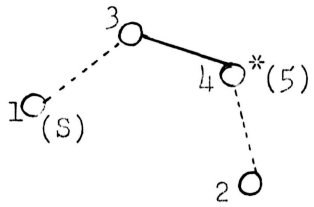
Standard Tree Types

TABLE 6

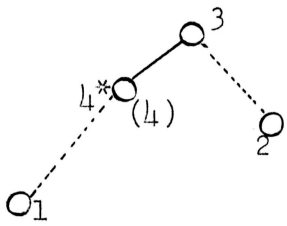
10.



11.



12.



13.

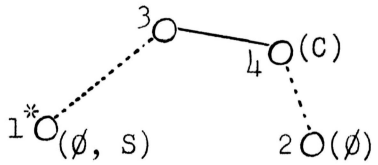


TABLE 6 (continued)

After the first dollar sign, the programmer must punch the tree specifications. If more than one tree specification is given, they are separated by commas. The tree specification consists of an unsigned decimal integer between 1 and 13 referring to a standard tree format from Table 6. The program supplies the dependency and syntactic relations. The tree type may be followed by a serial number specification in one of three forms:

(1) + assigns a serial number one greater than the last serial number assigned;

(2) * assigns a serial number equal to the last serial number assigned; and

(3) / followed by an integer assigns a serial number equal to the given integer.

No serial number need be specified.

If the data fields described above extend past column 72 of one card, they may be continued by placing a minus sign as the last nonblank character on the card, and punching five blanks followed by an asterisk in columns 1-6 of the next card; TRECND skips to column 7 of this next card when the minus sign is read.

Again, a card with ////// in columns 1-6 returns control to the supervisor of the criterion tree update.

Some Sample Criterion Trees: In the present dictionary, the stems for the words "DATA" and "INFORMATION" are classified under concepts 53 and 114 respectively; "RETRIEVAL" is in category 26.

Sample English sentence:

Information retrieval is useful.

MAKTRE format

```
INFRET                                    6
     1      X      (0026)                6
     2      1I     (0053,0114)           6
```

Sample English sentence:

Retrieving information is useful.

MAKTRE format

```
INFRET                                    7
     1      X                             7
     2      1D     (0026)(V)             7
     3      1I     (0053,0114)(Ø)         7
```

Sample English sentence:

Information is being retrieved.

MAKTRE format

```
INFRET                                          8
     1      X                                   8
     2      1I      (0053, 0114)(Ø, S)          8
     3      1D      (C)                         8
     4      3I      (0026)(Ø)                   8
```

All the above trees are written by the TRECND format card:

INFRET(26)/(53,114)$1/6,3+,4+

Statistical Phrase Dictionary Updating: File four of the library tape contains a set of concept n-tuples used in the phrase searching routine described in Part 5(A). This phrase searching procedure searches for a pair of words occurring in the same sentence. To this effect a file is required which contains a set of phrase names, up to six component concept numbers which make up the phrase, and a concept number associated with the whole phrase. Each phrase is placed on one BCD card, with columns 1-6 blank, the BCD name in columns 7-12, and the output concept number in 17-20. The component concept numbers are given in five column fields starting with column 21, right-adjusted within each field.

This deck of cards, sorted on the BCD identifier, is preceded by a control card which may be one of four types:

(1) REPLAC in columns 1-6, which writes the new cards onto B5, skipping past the old file on A6.

(2) IGNORE in columns 1-6, which is identical with REPLAC except that A6 is ignored.

(3) MERGE in columns 1-6, which adds these cards to those already on A6. However if a name appears which is the same as a card name already on A6, the program examines the concept number of the new card. If it is 9999, the phrase is deleted from the file. If it is not 9999, the new card replaces the old phrase. The new file is written on B5. and

(4) COPY in columns 1-6, copies the old file to B5 unchanged.

On any of these cards, if PRINT or PUNCH is placed in columns 7-12 a copy of the new file is obtained on the print or punch tapes, respectively.

Note: For every criterion tree there must be at least one statistical phrase with the same BCD identifier.

Hierarchy Updating: The concept hierarchy occupies the last file on the library tape. The hierarchy provides for structured relations between concepts, in which the concepts are nodes of one or more tree structures, as well as for looser "cross-reference" relations. Full facilities for expanding requests or documents in any direction are included in SMART, as described in Part 5(A).

Any number of tree structures, each containing any number of concepts, may be formed. Each concept, of course, can be included in only one tree. A concept may have any number of "brothers" or "sons" or "cross-references," but at most one "parent."

Control cards for the hierarchy should follow the update cards for the statistical phrase dictionary. The control cards are punched in the following format:

columns 1-6: a six-character alphabetic code;

columns 7-10: a concept number, specified by a right-justified unsigned decimal integer. The number must be less than 1,000. This field is called I;

columns 12-15: another concept number, also specified by a right-adjusted decimal integer, called J.

columns 17-20, 22-25, 27-30, 32-35, etc: may contain more concept numbers, called K, L, M, N, ...; and

columns 11, 16, 21, 26, etc., may contain commas or blanks.

The card is terminated by the first blank or zero concept number field. No continuation cards are allowed. The order of the hierarchy update cards is immaterial except for BYPASS, TAPE, and FINISH (q.v.).

The effect of the various six-character codes is:

INSERT I is inserted in the hierarchy as the son of J, with K, L, M, ... as cross-references if they are present. J, K, L, ... must already be in the hierarchy.

TOPMAN I is inserted in the hierarchy as a "root node," i.e., a node without a parent. If J, K, L, ... are present, they are entered as cross-references.

REFERS J, K, L, ... are listed as cross-references of I. This instruction eliminates the need for continuation cards.

OUTREF J, K, L, ... are deleted from the cross-reference list of I.

DELETE I is deleted from the hierarchy. Its sons become sons of its parent if I had a parent, or root nodes if I was a root node. The cross-references of I and all cross-references to I disappear. J, K, L, ... are ignored.

ATTACH I is removed from its current position in the hierarchy and reattached with all children and cross-references as a son of J, or as a root node if J is blank.

TAPE This must be the first card of the update deck if it is used at all. An old hierarchy is read in from tape A6 and updating begins with the tree structures on that tape. If TAPE is not used, A6 is ignored.

BYPASS All preceding control cards are processed. Normally, control cards are processed at the end of the update rather than as the control cards are read, so that the order of the input deck may be ignored. BYPASS interrupts this procedure and causes the immediate interpretation and execution of all update cards that have appeared. BYPASS is used to permit a sequence of operations such as insertion, deletion, and reinsertion of the same node. Normally, for example,

```
INSERT 475, 162
DELETE 475
INSERT 475, 380
```

is illegal because the same node is inserted twice. But

```
INSERT  475,  162
BYPASS
DELETE  475
INSERT  475,  380
```

is acceptable and has the same effect as

```
INSERT  475,  380
```

since the INSERT executed at the BYPASS instruction is canceled by the later DELETE.

FINISH — All hierarchy update cards are processed and the final hierarchy written on B5. If I = 2, the hierarchy is also listed on the print tape; if I = 1, the hierarchy is listed and also condensed internally before it is added to the library tape. SMART proceeds to chain link 1, provided that no serious errors have occurred in the updating.

Further details on hierarchy updating can be found in Section V of Information Storage and Retrieval, Report No. ISR-7.

5. Processing Options

It may be assumed at this point that SMART is read into core, and its chain links are set up on the tapes needed, with the library ready on B5. Most runs begin at this point; it is the position attained by loading the SMART program tape from B3.

The first thing required by SMART is a list of six-character identifiers for the concept numbers. Since this deck has three possible formats, its description is deferred to Part 5(B) to permit discussion of the processing specifications.

A. Specifications

SMART first demands a list of the options to be performed during the execution of the present computer run. The options are punched anywhere in columns 1-72 of any number of BCD cards. Specifications are separated by commas; blanks are completely ignored.

Many processing specifications exist in two forms, a long form and a short, two-letter abbreviation. In the discussion below, the two letter codes are given in parentheses after the full codes.

Lookup Specifications: The lookup is performed for each text introduced in English. Whatever dictionary is found on B5 will be used. Various options control the amount of printed output. These are:

ENGLISH TEXTS (ET)	prints each text as it is read in. The sentence numbers are printed in the right margin;
WORDS NOT FOUND (NF)	prints a list of words not found in the dictionary for each text. The word and sentence numbers in the text of each word not found are also printed;
THESHR n	the dictionary being used is the Harris thesaurus, version n. This specification only affects one identifying printout;
MAXCON nnn	the largest concept number in the dictionary is nnn;
THESNL n	the dictionary being used is the null thesaurus, version n.

Phrase Searching: SMART provides three methods of hunting for phrases in text. The first of these is a statistical method based on sentence co-occurrence. Words which consistently occur in the same sentences are grouped into clusters and counted as a unit. This grouping is performed by first constructing a table of related concepts (note that concepts not words are grouped since the phrase processing is performed after the lookup). The program produces a table of concept-sentence occurrences, and performs a correlation of this table to obtain term-term correlation values for each pair of terms. A cutoff is applied to reduce the resulting numeric values to logical (binary) ones, and the surviving links are placed in the term relation table. The relation list is treated as a connection matrix, and maximum connected subgraphs are extracted as clusters.

The second method involves a preassigned dictionary of word pairs or n-tuples (up to six). Whenever all members of a concept n-tuple occur at least m times in the same sentence, a record is made of the m occurrences of this phrase. The dictionary used for this purpose is the fourth file on tape B5, the "statistical phrase" dictionary. This dictionary contains a concept number associated with the whole phrase (as well as the concept numbers of all components), and the phrase concept is recorded for each occurrence of the phrase. Note by contrast that clusters detected by the program must be assigned imaginary concept numbers (since no dictionary is used) which can only match other occurrences of the same cluster, rather than possibly matching other words or hierarchical expansions, as is the case with phrases found in the phrase dictionary.

The third way of finding phrases also uses a prewritten dictionary, but this "criterion tree" dictionary contains a much more precise specification of each phrase. The criterion tree file (the third file on B5) contains a complete syntactic/semantic/structural specification of each phrase. Each component concept must have the proper semantic information, syntactic role, and must be properly related structurally to the rest of the phrase. A complete syntactic analysis of the text is performed in order to use the criterion tree file for phrase detection. The analyzer used is the Kuno Multiple-path Syntactic Analyzer. Only the first syntactic analysis of the several which the analyzer may produce is actually used, although all analyses may be printed. The Sussenguth graph-matcher (see Section VII of Information Storage and Retrieval, Report No. ISR-7) is then used to compare the dependency tree produced by the analyzer with the reference library of criterion trees on the tape. This process takes much more time than the previous (statistical) method; in particular, the time required for syntactic analysis may vary widely.

To perform these various phrase searching procedures, another lookup is necessary. The specifications required are described below:

Clustering:

CLUSTERING SEARCH (CS)	performs the clustering and alters the concept-vector of the documents accordingly;
WORD FREQUENCIES (WF)	prints the term-sentence matrix;
TERM CORRELATIONS (TC)	prints the term-term correlation matrix;
TERM RELATIONS (TR)	prints the list showing which terms are related to which other terms;

STATISTICAL CLUSTERS (SK)	prints the clusters detected by a CS search;
CORMD1 aaaaaa	the correlation method used in clustering terms is represented by aaaaaa = COS, OVLAP, ASYM, REDUCE. See Part 5(C);
CUTOF1 xxxx	cutoff for term-term correlation is 0.xxxx.

Search of Word n-tuple Dictionary:

PHRASE SEARCH (PS)	performs search of dictionary and adds detected phrases to concept vector;
STATISTICAL TREES (ST)	prints all detected phrases;
FRASES yzz	assigns a weight to the phrases found by PS (relative to concepts detected by normal lookup) equal to the original weight multiplied by y.zz (i.e. FRASES 100 corresponds to equal weighting).

Search of Criterion Tree Dictionary:

EXECUTE SYNTAX (ES)	performs a syntactic analysis of the text, and searches the criterion tree file for matching phrases; adds these to the concept vector;
SYNTACTIC ANALYSIS (SA)	prints the first syntactic analysis found by the analyzer for each sentence;
ALL ANALYSES (AA)	prints all analyses found by the analyzer for each sentence (only the first will be used in phrase searching);
NODE CORRESPONDENCES (NC)	prints a table showing the correspondences between sentence nodes and criterion tree nodes for each phrase found;
CRITERION TREES (CT)	prints a table showing which criterion trees were found in the text;

CRITES yzz	weights the criterion trees by y.zz when adding them to the concept vector (similar to FRASES yzz);
REPLACE BY SYNTAX (RS)	if both CS and ES are specified, the CS clusters are thrown away. This specification is useful for long texts. Since the system will select 1/5 of the text for syntactic analysis (analyzing long texts is very expensive), statistical clustering can be used to select the sentences to be analyzed; the results of this clustering can then be supplanted by those of the syntax.

Document Collection Options: Once all documents have been looked up and the phrase analyses performed, the next step is to gather the documents into a large matrix consisting of concept-vectors. Document vectors may also be punched out for future input with *LIST control cards.

A number of other options are needed at this point as well. For example, SMART can distinguish titles from the body of the text during analysis. The relative weights to be assigned to each part may be specified. SMART will also completely ignore weights if desired, treating the concept vectors as logical vectors. The following options are provided:

PUNCH DOCUMENT DATA (PU)	punches out looked-up documents on to binary cards. See Part 6(A);
LOGICAL VECTORS (LV)	ignores all weights or frequencies;
TITLES yzz	weights titles, relative to body, by y.zz (if the specification reads TITLES ONLY, the body of the text is ignored).
TEXTS PROCESSED (TP)	prints a page giving the identifiers of all documents read in this run.

MERGED CLUSTERING (MK)	when a CS search has been performed, fake concept numbers are assigned to the detected clusters. Different documents may include the same cluster with different fake concept numbers. The program rearranges the fake concept number assignment so that all assignments agree from document to document; the results of this rearrangement are printed;
PHRASE FREQUENCIES (PE)	the document-concept matrix is printed.

Statistical Expansion of Requests: The programs are capable of altering the concept vector of a request by using the statistical relations between concepts derived from an examination of the entire collection. The basic scheme is to prepare concept-concept correlations showing how strongly concepts tend to co-occur in the same documents within a document collection. Note that the correlation previously described correlates concepts on the basis of co-occurrence in the same sentence within one document. The present correlation method considers the whole document collection. The application of a cutoff to the correlations defines a list of "related" concepts for any given concept. The request vector may now be altered by adding to its concepts the related concepts. The options involved are as follows:

CONCEPT FREQUENCIES (CF)	prints a concordance of the collection by concept number, showing for any given concept, all documents in which that concept occurs;
CONCEPT CORRELATIONS (CC)	prints the concept-concept correlation matrix;

CONCEPT RELATIONS (CR)	prints the concept-concept relations list;
SMEAR CONCEPTS (SC)	expands the requests by the addition of related concepts from the list printed by CR. (SC may be given without CR, of course);
SMEARED VECTORS (SV)	prints request vectors before and after alteration;
CORMD2 aaaaa	correlation mode is aaaaa = COS, OVLAP, ASYM, REDUCE;
CUTOF2 xxxx	cutoff is 0.xxxx.

The following two options are not completely coded.

CLUSTER CONCEPTS (KC)	prepares and prints a list of concept clusters, i.e., groups of concepts which tend to co-occur;
SMEAR REPLACING (SR)	instead of adding related concepts to a request vector, condense the vector by replacing the concepts in it by concept clusters of which they are a part.

Hierarchical Alteration of Requests: The concept hierarchy (file 5 on the library tape) may be used for systematic alteration of request vectors. The hierarchy includes a structured tree of concepts, in which each concept is included with a clearly defined set of "sons," "brothers," and a "parent," as well as a set of "cross-references" providing for more general interconnections. The hierarchy processing options can enter into the request vector the "parent," "sons," "brothers," or "cross-references" of each included concept. These may be added to the request while retaining or deleting the original concepts.

It should be noted that the hierarchy is specified in terms of the concepts used in the thesaurus. If the thesaurus is changed in any major way, the hierarchy will have to be altered as well. The options used for hierarchical expansions are:

HIERARCHICAL EXPANSION (HE)	expands concept vectors by use of the hierarchy;
HIERARCHICAL VECTORS (HV)	prints vectors before and after expansions;
GOTREE bbbbb	the additional concepts are to be generated from a search specified by bbbbb;

bbbbb	type of search
STEM	parents
LEAF	children
FILIAL	brothers
REFER	cross-references.

REPLACE BY HIERARCHY (RH)	discards old concept-vector and keeps only the concepts obtained as a result of the search.
EXPAND ccccc	If ccccc = REQ, only requests will be altered; if ccccc = DOCS only nonrequests will be expanded; if ccccc = ALL all concept vectors will be changed.

Request Answering: After an adjusted set of concept vectors for the requests and the documents has been prepared, it becomes possible to correlate the requests against the documents, and produce a list of "related" documents for each request. These are called the "answers" to that request. SMART can also, if desired, correlate all documents against all other documents, producing a complete document-document relation list and even document clusters.

The options involved in this processing are:

ANSWER REQUESTS (AR)	prints a list of answers to the requests;
REQUEST CORRELATIONS (RC)	prints the correlations of each request with each document. The correlation list is printed twice, once in increasing document order and once in decreasing correlation order. It is followed by a histogram showing the distribution of correlations. Zero correlations are not printed but are omitted from the lists;
DOCUMENT CORRELATIONS (DC)	prints the correlations of each document with each other document. Note that each correlation is still printed twice, so that for n documents $2n^2$ correlations are produced. 112 correlations are printed per page;
DOCUMENT RELATIONS (DR)	prints the document relations list. AR prints that portion of the list which involves the requests;
CORMD3 aaaaa	specifies correlation mode aaaaa = COS, OVLAP, ASYM, REDUCE;
CUTOF3 xxxx	cutoff is 0.xxxx.

The following specification is not completely coded:

DOCUMENT EXPANSION (DE)	adds to the list of answers for each request all documents related to any document in the list of answers;
CLUSTER DOCUMENTS (KD)	clusters the document collections and prints the results.

Other Specifications:

FORMAT ttttt	where ttttt = MEDIUM, BIG, or JUMBO. Selects among three output formats for

the answers. MEDIUM provides the most compact output, JUMBO the most detailed output, and BIG provides an intermediate amount of output. Formats BIG and JUMBO require chain link 8;

SCORES sss — where sss = YES or NO. Decides whether to call the automatic evaluation program (Part 6(C)).

B. Names Table

SMART can accept a list of six-character identifiers for the concept numbers, used in printing references to the concept numbers. This list must immediately precede the processing specifications (Part 5(A)) and follow the update cards (if any). If no updating is performed, the names table is at the beginning of the data deck.

There are three formats for the names table:

(1) It may be omitted entirely. SMART represents each concept number by its decimal equivalent in this case.

(2) It may be submitted in BCD form. Each identifier is placed on a separate card, which contains *NAMES in columns 1-6, the six-character identifier in columns 7-12, and an optional integer beginning in column 13 and terminated by a blank. The identifier is assigned to the concept number specified by the optional integer, if punched. If column 13 is blank, the identifier is assigned to the concept number one greater than the concept used for the preceding card. The first *NAMES card is understood to refer to concept 1 if no concept number is punched. After all identifiers are specified, a card with ****** in columns 1-6 should be submitted to end the table.

(3) It may be submitted in binary form. When SMART reads a names-table in BCD form, it punches an equivalent table on binary cards. This binary deck is one twenty-fifth the size of the BCD deck and can be used, exactly as punched, in place of the BCD deck described above. The first card of this table is a BCD card containing an octal integer in columns 1-3. This integer is the number of cards in the remainder of the deck. Further cards are binary and have 4-7-9 punches in column 1.

C. Correlation Algorithms

There are four correlation methods for comparing two vectors $\underline{x}$ and $\underline{y}$. They are:

(1) CØS

$$r_{xy} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}}$$

(2) ØVLAP

$$r_{xy} = \frac{\sum_i \min(x_i, y_i)}{\min\left(\sum_i x_i, \sum_i y_i\right)}$$

(3) ASYM

$$r_{xy} = \frac{\sum_i \min(x_i, y_i)}{\sum_i x_i} \neq r_{yx}$$

(4) REDUCE

$$r_{xy} = \frac{\sum_i x_i y_i}{\sqrt{\left(\sum_i x_i^2\right)\left(\sum_{x_i y_i \neq 0} y_i^2 + \left(\frac{n_x - n_{xy}}{n_y - n_{xy}}\right)\left(\sum_{x_i y_i = 0} y_i^2\right)\right)}} \neq r_{yx}$$

$$n_x = \sum_i \eta^+(x_i) \quad \text{(i.e., the number of nonzero elements in } \underline{x})$$

$$n_y = \sum_i \eta^+(y_i)$$

$$n_{xy} = \sum_i \eta^+(x_i y_i) \qquad \eta^+(x) = 0 \quad x \leq 0$$
$$= 1 \quad x > 0$$

6. Data

Data provided for SMART are normally documents, and search requests. Documents or requests may be submitted either as English text, or as looked-up binary decks punched by the PU specification. In general it is preferable to use looked-up decks when possible, because those load much faster and take less space in core (since fewer programs are needed to process them.) Furthermore, looked-up documents may be read from B3 as well as from A2. However, all changes in the thesaurus and all alterations in phrase searching procedures require new lookups from the original text.

The documents are introduced by means of SMART control cards. Control cards to SMART following the specifications (see Part 5) are BCD

cards with asterisks in column 1. They are the only cards which may have an asterisk in column 1. Four characters indicating the type of control card are contained in columns 2-5. Column 6 is blank. If the control card refers to a document, the document is identified by the contents of columns 7-18. All other columns of the control cards are ignored.

A. Control Cards which Introduce Documents

To introduce a document into the SMART system three control cards may be used. These are *TEXT, *FIND, and *LIST. On each of these cards the document name must be given in columns 7-18. SMART always refers to the document by this name; no two documents may have the same name, and no document should be given a name consisting of twelve blanks or twelve zeros.

*TEXT and *FIND cards precede English texts. A text prceded by a *FIND is treated as a request; a text preceded by *TEXT may be a request, but need not be. The text begins on the card following the control card and continues on as many cards as needed, subject to a limit of 1,000 English words.

Format of English Texts: Basically, input text to SMART resembles typescript. For example, text may be punched anywhere in columns 1-72 of any number of cards. Any number of consecutive blanks are equivalent to one blank. A space is assumed between column 72 and one card and column 1 of the next card.

Major differences from typescript are as follows:

(1) * in column 1 or $$ anywhere are end-of-text signals. These indicators should not be used unless an "end-of-text" indication is intended. Normally, the * on the control card following the text suffices to signal its end.

(2) Periods not preceded by blanks are taken to be parts of abbreviations. Thus, a period meant to indicate end-of-sentence should be preceded by a blank.

(3) SMART provides for the inclusion of up to 355 characters of identification for each text. This identification is used in the FORMAT JUMBO output option. This identification should be punched at the beginning of the text on cards with a single $ in column 1. If a text begins with cards with $ in column 1, SMART ignores their contents except that the first five such cards (or fewer) are collected as BCD identification. The text proper is assumed to begin on the first card without a $ in column 1 and any further cards with $ in column 1 are treated as part of the text. The identification may be omitted; in this case, the first card of the text must <u>not</u> have a $ in column 1.

(4) The convention used to hyphenate a word between two cards is to place a minus sign (11-punch) followed by a blank at the end of the first part of the word. SMART ignores the remainder of the card, throws away the minus sign, and continues from column 1 of the next card. Note that this rule makes the following construction illegal: "sub- and super-scripts." Also note that normally hyphenated words which are broken between two lines at the hyphen must have a double minus sign to be properly recognized.

(5) Hardware restrictions require that only upper-case letters be used. Special characters are treated as follows:

?	.QUE (preceded by a space).
!	.EP (preceded by a space).
"	/ (for both open and close quotes. There should be no space before a close quote and no space after an open quote. Similarly, parentheses and commas are spaced normally).
–	–DASH (11-punch minus sign followed by the word DASH. Hyphens not at the end of a line are treated normally; thus, "co-operate.")
'	' (8-4-punch, even if keypunch prints a minus sign).
;	, . (not preceded by a space).
:	.. (preceded by a space).

Format of Binary Documents for *LIST cards: Binary documents are obtained by the use of the PU specification. For each document looked up, the computer punches a binary deck which can be submitted in place of the text in future runs, so long as the thesaurus and the phrase searching specifications remain the same. The format of these binary decks is as follows:

(1) Two flip cards showing the document name, six characters per card, in large letters. This is an identification for the user, and these cards should be thrown out or placed after the *LIST card for proper processing by SMART, since they contain no information of any use to it. These are binary cards.

(2) A *LIST card usable for submitting the binary deck. This is a BCD card obtained by replacing the *TEXT on the original control card used to submit the text by *LIST, and adding the date in columns 73-80. Other parts of the *TEXT card remain unchanged.

(3) The actual binary document on a set of serialized, checksummed binary cards with 5-7-9 punches in column 1. The first eight characters of the document name are punched in columns 73-80 (in BCD). The

serialization is in binary in column 3. If a checksum error occurs (a very unlikely possibility,) the deck should be thrown out and the document looked up again. There is no ignore checksum punch. Any identification punched in the original text (Part 6(A)) is saved in these decks.

B. Control Cards which do Not Introduce Documents

The remaining control cards do not introduce documents. They must therefore be followed by other control cards. Although they do not precede text, two of the cards do refer to documents. These control cards are:

(1) *LIKE A document name that refers to an already introduced document is given in columns 7-18. This document is marked as a request.

(2) *RAND A pseudo document with a randomly generated concept vector is produced and given the name specified in columns 7-18.

(3) *ONLY No more English text is being submitted; i.e., no more *TEXT or *FIND control cards will appear. Use of this control card after the last English text saves core by permitting SMART to erase the programs needed to process English.

(4) *TAPE SMART tape (B3) is used and spaced to the data section, and control cards and data are taken up in order. This instruction also prohibits further *TEXTS or *FINDS, as does *ONLY, thus text may not be put on B3. The effect of this card is to switch the input tape from A2 to B3. The card is not legal if SMART is currently reading from B3.

(5) *FILE *FILE cards may precede a *TAPE control card to permit access to different files on B3. Each card causes SMART to space one file into the data section before reading; by using enough such cards, any file may be reached. *TEXT and *FIND cards

may not appear between *FILE and *TAPE, since they cause a rewind of B3. *FILE occurring as a control card on B3 will cause SMART to skip beyond the next EOF on B3 and continue reading.

(6) *BACK — This restores the input tape to A2. It should be the last card of each data file on B3. B3 is also rewound. If *BACK appears on A2, the rewind operation will still be executed and the card otherwise ignored.

(7) *STOP — All documents have been submitted. SMART can now perform all processing except evaluation (Part 6(C)) can follow this card.

(8) *TIME — SMART finds out, determines the current time, (printer clock, RPQ 78054 required) and prints a message.

(9) *NOTE — Ignored; used for putting comments on the print tape.

C. Evaluation

If automatic evaluation has been requested (SCORES YES) SMART must be given a list of the documents believed to be relevant to each request for purposes of comparison. This list follows the *STOP card. Its format is as follows:

(1) A card containing a decimal integer in columns 1-3 giving the number of requests this run.

(2) For each request, a card giving the name of the request in columns 1-12 and the number of documents judged relevant in columns 13-16.

(3) Following this card (Part 6(C)) one card for each relevant document, giving its name in columns 1-12.

(4) (2) and (3) must be repeated as many times as indicated by (1).

The requests must be in the order in which they were submitted to SMART.

This produces, for each request, a page showing the top fifteen documents in correlation order, the relevant documents, their correlations, and their rank orders, and a set of recall and precision measures for this request.

7. A Sample Input Deck

Note: Normal typing represents comments; UPPER CASE UNDERLINED represents actual input cards; lower case underlined represents binary input decks which cannot be typed verbatim.

* JOB,630201,5MIN,5000,LESK RUN SMART

* XEQ

These are the normal FORTRAN monitor control cards needed to submit a run.

* PLEASE MOUNT TAPES ...

* SMART PROGRAM/DATA TAPE SALTON 1 ON B3 RING OUT

* SMART LIBRARY TAPE SALTON 12 ON B5 RING OUT

* PAUSE (THANK YOU)

These cards permit the operator to check the tape mounting.

subroutine tapld (two binary cards)

This program, when executed, loads the remainder of the SMART system from tape B3.

* DATA

The FORTRAN monitor begins execution of SMART. All further cards are read by SMART, not by the monitor.

names table (see Part 5(B)) submitted as 22 binary cards

This table permits SMART to substitute alphabetic identifiers for the concept numbers in printouts. The first of the 22 cards is really octal BCD, not binary.

ENGLISH TEXTS, WORDS NOT FOUND, THESHR 2, MAXCON 511, TEXTS PROCESSED, TP, T P TEX-TS PROCES SE D, PF HIERARCHICAL EXPANSION, HIERARCHICAL VECTORS, GOTREE STEM, REPLACE BY HIERARCHY, ANSWER REQUESTS, REQUEST CORRELATIONS, CORMD3 COS, CUTOF3 3500, FORMAT JUMBO, SCORES YES

These specification cards produce the following output:

(1) Each English text is printed as it is read in.

(2) The words in each text that are not found in the dictionary are printed.

(3) The dictionary being used is the Harris thesaurus, version two (this specification is used only for identification).

(4) There are not more than 511 different concepts in the thesaurus.

(5) The list of texts processed in this run is printed. This specification is given in four ways. All are acceptable and the redundant ones are ignored.

(6) The concept vector for each text is printed.

(7) A hierarchical alteration of the requests is performed.

(8) The vectors are printed before and after alteration.

(9) The direction of the hierarchical expansion is toward the parents of the concepts present in the request vector.

(10) The new concepts replace the concepts in the old vector, rather than being added to them.

(11) Requests are answered.

(12) The correlations of the requests are printed.

(13) The correlation method used is the cosine correlation.

(14) The cutoff is set at 0.3500.

(15) The most detailed format for the answers is used.

(16) The results of the run are evaluated automatically.

*NOTE USING HIERARCHY OF AUGUST 3, 1964

This comment is printed and ignored.

*TIME

The time when this card is processed is printed.

*FIND PATTERN RECG A REQUEST ABOUT PATTERN RECOGNITION

TELL ME ABOUT PATTERN RECOGNITION . HOW ARE TWO-DIMENSIONAL PATTERNS SCANNED AND ENCODED FOR A COMPUTER .QUE WHAT PROCEDURES ARE USED TO DETECT AND IDENTIFY GEOMETRICAL SHAPES, AND TO RECOGNIZE ALPHABETIC CHARACTERS .QUE

A request named PATTERN RECG is submitted in English.

*LIST MORSE CODE

a deck of binary cards (six cards, punched when this document was looked up in the dictionary) is submitted at this time

The programmer has already looked-up MORSE CODE in the thesaurus and chooses to submit the binary deck to save time. This document is accepted as if it were a text.

*LIKE MORSE CODE

MORSE CODE is marked as a request.

*TAPE

SMART goes to tape B3 and begins reading control cards from the data section. The first file of the data section contains 405 looked-up abstracts. It appears as follows:

```
*LIST 1A COMPUTER ORIENTED TOWARD SPATIAL
     binary deck for document 1
*LIST 2 MICRO-PROGRAMMING
     binary deck for document 2
*LIST 3
       .
       .
       .
*LIST 405 COMPUTER POWER .. A PUBLIC UTILITY  .QUE
    binary deck for document 405
*BACK
    return to A2 input
```

*TIME

Again, the time is requested.

*STOP

All documents have been submitted; the computer processes all remaining specifications except SCORES.

002

There were two requests in this run. This card begins the evaluation data for scoring.

PATTERN RECG 10

1A COMPUTER

39AUTOMATIC

54AN AUTOMAT

163A COMPUTE

205 AUTOMATIC

224A METHOD

314THE MECHA

350PATTERN R

351A SYSTEM

353THE DESIG

All the relevant documents for PATTERN RECG are listed. The computer evaluates the results of this processing for PATTERN RECG using this list as the standard.

MORSE CODE 2

305MACHINE R

394THE MORSE

The relevance data for MORSE CODE now follows. This is the end of the input deck. A total of 69 input cards are submitted, 39 BCD cards and 30 binary cards.

end of file

8. Miscellaneous

SMART is constantly being altered and revised. Thus, writeups may differ. The present writeup postdates Information Storage and Retrieval, Report No. ISR-7. No guarantee can, however, be made as to the proper functioning of the programs.

A. Size Limits

Thesaurus: 1,000 concepts. 2,500 stems.

Texts: Less than 1,000 words. About 450 100-word texts can be processed in one run.

B. Timing

Lookup of a document: 10 seconds.

Syntax: Varies greatly. 3-4 minutes per document is probably the maximum profitable time to be allowed.

Read in 400 looked-up texts: 40 seconds.

Correlate, print full answers and evaluate a request: About 10 seconds for a collection of 400 documents.

Hierarchical expansions: Double correlation time.

Statistical expansions: No valid time estimates available. An estimate may be 10 minutes for a full concept-concept correlation with all correlations printed.

C. THES

A program is available which generates a null thesaurus from English text. This program also prepares a frequency count of the text. (THES) is independent of SMART and a separate description is available in Part 3, Section XI of Information Storage and Retrieval, Report No. ISR-7.

APPENDIX 1

SMART TAPES

A1 FMS tape. This tape is rewound at start of job and used for CALL EXIT at end of job.

A2 Card input. 14-word BCD and 28-word binary records.

A3 Off-line print. BCD records, about 340 characters long.

A4 All of this tape except for the last record is used as identification for the options FORMAT BIG and FORMAT JUMBO. See Parts 5(A), and 6(A). The last record is used to save core storage while lookups are being performed. Since this record is 20,000 words long, redundancy checks are somewhat more probable than usual and dirty tapes or bad tape drives should be avoided for this unit. 12-, 60-, and 20,000 word binary records are used. At the beginning of the run they are used as a chain tape.

A5 Kuno condensed grammar 6C.

A6 Output from lookup routine. Used for preparing input tape to Kuno analyzer. Binary records, ranging from 9 to 900 words long. Then output from analyzer; 22-word BCD records. In update mode contains the old library.

B1 Used by the loader as scratch. Used by the hierarchy update and the criterion tree routines as scratch. Used by the automatic evaluation program to obtain the correlation data. Used by the Kuno analyzer as a binary sentence tape. Mode and length of records varies.

B2 Chain link 12 if submitted; DUMP-PDUMP scratch (not usually used); binary analysis tape for Kuno analyzer. Mode and length of records varies.

B3 Chain tape for links 2-9 if running from cards; Part 4(A) describes its use as a program/data tape.

B4 Off-line punch. 28-word binary records.

B5 Library tape. Binary, five files. See Table 3.

B6 Augmented text tape for input to Kuno analyzer. 14-word BCD records.