INFORMATION AND CONTROL 5, 368–384 (1962)

The Degarbler—A Program for Correcting Machine-Read Morse Code

CONSTANCE K. McELWAIN AND MARTHA B. EVENS

Lincoln Laboratory, [*] *Massachusetts Institute of Technology, Lexington, Massachusetts*

An IBM 7090 program automatically corrects garbled samples of English text. The garbles are intended to resemble those caused by Morse Code transmissions.

The program has access to a vocabulary and a table of the Morse Code equivalents of the English alphabet.

The correction rate on text in which 0–10% of the characters have been subjected to Morse Code garbles is about 70%. The apparent improvement in intelligibility is very marked.

INTRODUCTION

Several years ago a special purpose digital computer called MAUDE [Morse AUtomatic DEcoder (Gold, 1959)] was built to decode hand-sent Morse Code signals. Since MAUDE was designed to handle all types of Morse transmission, it did not use any information that was dependent on the language of the message. Although MAUDE operates surprisingly well it still makes many mistakes. Two questions arose. First, could certain known linguistic constraints be used to improve its output? Second, how effective in the reconstruction of mutilated English text is such rudimentary context as the knowledge of vocabulary? The Degarbler represents an attempt to answer these questions. It is an IBM 7090 program which at least partly corrects English text that has been typically corrupted in decoding by MAUDE. The program is limited to text which if error-free would contain only alphabetic characters and spaces. (Hyphens and apostrophes must have been eliminated and the remaining punctuation and the numerals must have been spelled out.)

The Degarbler starts with certain information; inter alia, the text

[*] Operated with support from the U. S. Army, Navy and Air Force.

to be corrected. The first section describes that information. The second section describes in some detail the correction procedures. The third section summarizes the tests run with actual texts.

BASIC INFORMATION

The Degarbler is supplied initially with a vocabulary table containing all of the words occurring in the messages to be processed, and with a table of the Morse Code equivalents of the alphabet and the numerals. An auxiliary program uses these two tables to construct the following lists used in the Degarble program itself.

1. Vocabulary. Each word in the vocabulary list is tagged to show if it is dockable (that is, if it remains in the vocabulary without its final letter) and if it is a subword (that is, if it forms part of another word in the vocabulary). For example, area (are) and forty (fort) would be tagged as dockable words; on (*on*slaught, c*on*stant, automati*on*) would be tagged as a subword.

2. Digrams. Each digram occurring in the vocabulary (i.e., set of two consecutive letters in a word) is entered on the digram list, along with a sublist of all the words which contain it. The words on each sublist are arranged in order of decreasing number of marks (i.e., dots and dashes) contained in their Morse Code equivalents. Words containing an equal number of marks are ordered by decreasing number of dashes. [The number of marks and the number of dashes is computed from the list of Morse Code equivalents (see Table 1)].

3. Trigrams. A list of trigrams (i.e., sets of three consecutive letters in a word) is constructed in the same way as the digram list.

TABLE I

MORSE ALPHABET

A	·–	J	·–––	S	···	1	·––––
B	–···	K	–·–	T	–	2	··–––
C	–·–·	L	·–··	U	··–	3	···––
D	–··	M	––	V	···–	4	····–
E	·	N	–·	W	·––	5	·····
F	··–·	O	–––	X	–··–	6	–····
G	––·	P	·––·	Y	–·––	7	––···
H	····	Q	––·–	Z	––··	8	–––··
I	··	R	·–·			9	––––·
						0	–––––

CORRECTING A MESSAGE

Preliminary Processing

As a message is read into the computer the Degarbler eliminates the word spaces and stores the message as a string of letters. The original word space information is recorded for use by the letter correction routines. Since all the numerals in the original text were spelled out, any numeral found in the message must be a garble. If the read-in routine encounters a 6 ("6" is −···· in Morse Code) it is replaced with the letters TH. ("T" is −, "H" is ····). A 9 ("9" is −−−−·) is changed to ON ("O" is −−−, "N" is −·). Other numerals are not altered. The program notes the location of these substitutions. Later it will consider other solutions if these have not been successful.

Determination of Word Spaces

The space routines insert word spaces in the letter string[1] and tag as "dangling" any letter which cannot be incorporated into a word. To do this the space routines start at the beginning of the letter string and match the letters against the vocabulary. The longest initial string of letters which forms a word is selected. Example: Given the string ADDITIONALENERGY the program chooses ADDITIONAL rather than ADDITION or ADD or A. The matching process resumes with the first unused letter and is repeated over and over until the remaining string begins with a sequence of letters which does not form a word. Then an attempt is made to change an earlier decision. If one of the previous words is dockable, beginning a vocabulary search with its last letter may resolve the current problem. If the attempt fails the first letter of the unsolved sequence is designated a dangling letter and the search resumes with the following letter. Example: given the letter string FIELDSITESNEAR the space routines initially select FIELDS IT but since the remaining string ESNEAR does not begin with a word, previous decisions are analyzed. IT is a dockable word so the program tries FIELDS I as a partial solution. The remaining letter string

[1] In trying to determine word spaces in the letter string rather than trying to correct the spacing that exists in the input message we are undoubtedly ignoring useful information because the input message rarely has more than 20% of the word spaces in error. However, we were unable to derive a set of rules which made adequate use of the original word space information. At present the original word space information is used only as one of several factors which determine criteria in the letter correction routines.

TESNEAR does not begin with a word so this solution is rejected. Since IT is also a subword it is capable of being incorporated into a longer word so FIELDS is examined. FIELDS is docked and a vocabulary search started with the docked S. *This produces the solution FIELD SITES NEAR.*

Error-Sequences

When the end of the letter string is reached the message will consist of words interspersed with dangling letters. The presence of a dangling letter indicates that there is an error in the vicinity. The boundaries of an Error-sequence around each dangling letter or group of dangling letters are now defined. A word in the vicinity of a dangling letter will be retained in the Error-sequence if it might need to be changed to solve the garble. In general, a questionable word (i.e., a dockable word or a subword) which abuts a dangling letter is retained in the Error-sequence which is set up around the dangling letter.

Rules for Determining an Error-Sequence

In the following rules a long word is defined as a word which contains at least three letters. An acceptable word is a long word which is neither dockable nor a subword.

A. Search through the text until a dangling letter is found.

B. To determine the beginning of the Error-sequence back up checking each word.

1. Include in the Error-sequence a long dockable word which is not a subword, and consider it the beginning of the sequence.

2. Do not include in the Error-sequence an acceptable word or a long subword which cannot begin another word. The Error-sequence will begin immediately after such a word.

3. Include in the Error-sequence a word not covered by (1) or (2) and continue.

C. To determine the end of the Error-sequence examine the text following the dangling letter.

1. Do not add to the sequence an acceptable word, a long subword which cannot end a word, or a long dockable word. The Error-sequence will end immediately before such a word.

2. Add to the sequence a word not contained in (1) or a dangling letter and continue checking.

All the errors should now be included in Error-sequences. These are the only areas of the text that are processed by the correction routines. The second paragraph in each of the examples below corresponds to a print-out of a message at this point. The Error-sequences are enclosed in parentheses.

The Correction Routines

Each Error-sequence is processed separately. Words which have a trigram or digram in common with the Error-sequence are appraised as possible solutions. If a word satisfies the correction criteria it is substituted for the appropriate section of the Error-sequence.

The correction routines start at the beginning of an Error-sequence and examine each combination of letters to find the first occurrence of a trigram or digram which is in the vocabulary. When a trigram or digram is found the routines analyze the list of words which contain it. If the program has a choice it will analyze the list of words suggested by a trigram. (The words which contain a digram will be checked only if the digram and the letter following it do not form a trigram in the vocabulary or if the trigram they form has failed to provide a solution.)

Words on a trigram or digram list are evaluated in order until a solution is found or until the list is exhausted. When a list is exhausted, the program moves onto the next trigram or digram and repeats the process.

At present the Degarbler is not sophisticated enough to be able to solve every garbled word that occurs in the messages. It can correct a word only if the word contains one of the following types of errors.

Error Classification

(See Morse Code Table—Table I)

TYPE I — A letter space mistaken for an intraletter space[2] or vice versa. (This causes a letter to be split into two letters or two letters to be combined to one letter.)
EX: *A* garbled to *ET*; *AN* garbled to *P*

TYPE II — A dot instead of a dash or vice versa.
EX: *A* garbled to *I*; *A* garbled to *M*

TYPE III — An extra or missing dot.
EX: *A* garbled to *U*; *A* garbled to *T*; *AN* garbled to *AEN* or *MEN* garbled to *MN*.

TYPE IV — An extra or missing dash.

[2] See definitions Table V.

EX: *A* garbled to *E*; *A* garbled to *K*; *THE* garbled to *HE*; *HER* garbled to *HETR*.

TYPE V Any combination of two of the previous errors as long as at least one error is of TYPE I or II.
EX: *A* to *ETT* (TYPE I AND TYPE IV)
EX: *ANI* to *PEE* (2 TYPE I ERRORS)

A garbled version of a word which contains any other combination of errors is called a TYPE VI word. The Degarbler has not been programmed to correct such a word. Notice that although the program may make two changes in a letter string to map it into a word (TYPE V), only one of the changes may alter the number of marks in the string.

Choosing a Letter String

In order to evaluate a word as a possible correction, it is necessary to select a string of letters in the Error-sequence which might comprise a garbled version of the word. In selecting this string of letters the routines take into account the restrictions imposed by the types of errors which the program can correct. Often it is possible to select more than one string of letters which might be a garbled version of a word. In this case all such strings are evaluated.

For example, consider the Error-sequence (T H E E *I* T I M A T E D E N E R *R* Y) a garbled version of THE ESTIMATED ENERGY. Examining the words which contain the first few digrams and trigrams will not produce a correction. When the program considers words containing the TIM trigram the word ESTIMATED will be evaluated as a solution. There are three possible letter strings in this Error-sequence which could be a garbled version of the word ESTIMATED.

1. EITIMATED contains one less mark than the word ESTIMATED
2. EEITIMATED contains the same number of marks as word ESTIMATED
3. EEITIMATEDE contains one more mark than the word ESTIMATED

Any other string which could be chosen would differ from the word ESTIMATED by more than one mark.[3] The program knows it cannot correct such a string so it doesn't bother to analyze it.

[3] EITIMATEDE can be ruled out because it differs by one mark on each side of TIM trigram, i.e., would require two changes of the number of MARKS to correct it.

Processing a Letter String

Each letter string is subjected to a series of checks which are intended to inhibit the program from making false corrections. First each string is checked to see if it is an exact match for the word. (The word could have been retained in the Error-sequence because it was dockable or a subword, etc.) If a match is found, the string will be accepted and the digram search will be resumed with the letters that follow. If not, each letter string is checked to be sure that it contains at least one dangling letter. Any string which does not is discarded because each letter in it has already been incorporated into a word by the vocabulary search.[4] Next the program makes up the Morse Code pattern for the vocabulary word and for the letter strings still under consideration. It ascertains for each letter string whether one of the five types of allowable changes would map it into the vocabulary word.

Value of a Letter String

A letter string that can be mapped into a word is considered to have a Value relative to that word. In the early versions of the Degarbler the Value of a letter string was defined as the number of marks contained in the Morse Code equivalents of the letters in the string. However, several adjustments were eventually made to the definition in order to attempt to inhibit some of the Degarbler's mistakes. Currently the Value of a letter string with respect to a particular vocabulary word is equal to the number of marks in the Morse Code equivalent of the word, plus the number of letters in the word in excess of four, minus twice the number of word or letter spaces in the original text which this change would reverse, plus a bonus of three if the change would completely solve the Error-sequence. The Value of each string is compared with an empirically determined threshold for the type of error involved. If the Value is greater than or equal to the threshold for the type of error involved, the change can be made. If more than one string of letters has a Value greater than or equal to its threshold, the shortest string is chosen. Occasionally a one mark letter (an E or a T) is left over but this is prefer-

[4] This check helps to inhibit the Degarbler from changing a correct word. EX: If DIVIDES is included in an Error-sequence this rule will prevent its being changed to DIVIDED (which precedes DIVIDES on the DIV trigram list). We cannot automatically assume a word such as DIVIDES is correct without checking the DIV trigram list because the Error-sequence might have contained SSDDIVIDES—a TYPE V garble for SUBDIVIDES.

able to the program's occasionally stealing an E or a T from an adjacent word.

For example, if a section of the original message had been

WH EN6E EITIMAT EDEN ERRY

the space routines would have set up the following Error-sequence:

WHEN (*TH*EEITIMATEDENERRY)

(Note the automatic 6 → TH change made by the Read-in routine). When the program analyzes ESTIMATED because it contains the trigram TIM, it will set up the following table:

	Possible strings of letters	Type of correction required	Threshold for this type of correction	Value of string
I	EITIMATED	TYPE III	16	17
II	EEITIMATED	TYPE I	10	13
III	EEITIMATEDE	TYPE V	20	13

Two of the strings have a Value greater than the corresponding threshold. String I (the shorter string) is accepted. When a string of letters is accepted it is deleted from the Error-sequence and replaced with a copy of the vocabulary word. The trigram search is resumed with the letters ENE. Notice that if string II had been chosen, a letter would have been stolen from the previous word. The program is allowed to consider using a letter or letters that are currently part of a previous word because the fact that the program has retained a word in an Error-sequence means that there is doubt about the word being correct. For example, consider the Error-sequence (HEATEDITTED). When the HEA trigram list is investigated, HEATED will be assumed to be correct. Then the program will consider the ITT trigram list. The Error-sequence will not be solved unless the program is allowed to use the final ED of HEATED when it is making up strings of letters to be mapped into the word EMITTED.

6 → TH or 9 → ON Substitutions

A letter string which contains a 6 → TH or a 9 → ON substitution can still have a Type V change made in it so occasionally a string is corrected which originally contained three errors.

When a word is tried as a correction and fails, the letter strings are checked to see if any of them contains a 6 → TH or a 9 → ON substitution. If so, and if the TH or ON is not part of the trigram or digram which caused the word to be considered, the 6 or 9 is restored and the word is reanalyzed as a solution. If this procedure also fails, the TH or ON is resubstituted and the program continues. For example, if the word DICTATE had originally been garbled to 6KTATE, it would be changed to THKTATE by the read-in program. When DICTATE is tried as a solution because of the common TAT trigram, it would fail because three changes are required. Then the program would try DICTATE as a solution to the letter string 6KTATE and succeed since this string contains only a TYPE V error.

Further Correction Procedure

When the correction routines have processed all the Error-sequences, many of the garbles will have been corrected. Paragraph three in each of the examples corresponds to a message printed out at this point.

Since most of the messages contain garbles the Degarbler is not programmed to correct, and since the thresholds on the first pass are too high to allow most 3- and 4-letter strings to be altered, it is expected that there will still be errors in the text. Therefore the thresholds are lowered and each remaining Error-sequence is again processed by the correction routines. This is the last attempt at correction and the results are shown in paragraph four of the examples. In Example I which has only 10 words in error (3%), lowering the thresholds and reprocessing produced no further corrections.

Example I

52 Space errors	10 Words in error
(15%)	(3%)

Input Message

F OR ENAAMPLE COM MA THE EAR TH LOSES S OME OF ITSH EA T TOSFA CE DAYA NDNIGHT COMMAA ND IS SA I D TO7UOTATIO N MARK COOL BYRA6 ATTON QUOTATION MA RK PERIOD OFC OUR SE COMM A UHEN THE SUN IS SHINING CLEAR LY COMMATHE IL LUMIN ATED SIDE OF THE EAR6 IS GAIN I NG MORE ENER GY F ROM 6E SU N THA N ITIS LOSING TOSP A C E PERIO D THERA TE OFRADIAT ION INCRE ASES AATHE FOURTH POW ER OF

THE TE MPER A TTRE EXPRESSEGIN THEA KSOLU TE
SCA LESEM ICO LON

Message After Initial Processing

Error-Sequences in Parentheses

(FORENAAMPLE) COMMA THE EARTH LOSES SOME OF (ITSHEATTOSFACE) DAY AND NIGHT COMMA AND IS SAID (TO7UOTATION) MARK (COOLBYRA*TH*^{12}ATTON) QUOTATION MARK PERIOD OF COURSE COMMA (UHEN) THE SUN IS SHINING CLEARLY COMMA THE ILLUMINATED SIDE OF THE EAR*TH*^{6} IS GAINING MORE ENERGY FROM *TH*^{6}E SUN THAN IT IS LOSING TO SPACE PERIOD THE RATE OF RADIATION INCREASES *A AT*^{11} HE FOURTH POWER OF (THETEMPERATTREEXPRESSEGINTHEAKSOLUTE) SCALE SEMICOLON

Results of First Letter Correction Pass

Thresholds: TYPE I 10, TYPE II 12, TYPE III 16, TYPE IV 14, TYPE V 20

FOR *EXAMPLE*^{1} COMMA THE EARTH LOSES SOME OF ITS HEAT TO *SPACE*^{2} DAY AND NIGHT COMMA AND IS SAID TO *QUOTATION*^{3} MARK COOL BY *RADIATION*^{13} QUOTA TION MARK PERIOD OF COURSE COMMA *WHEN*^{2} THE SUN IS SHINING CLEARLY COMMA THE ILLUMINATED SIDE OF THE EARTH IS GAINING MORE ENERGY FROM THE SUN THAN IT IS LOSING TO SPACE PERIOD THE RATE OF RADIATION INCREASES *A AT*^{11} HE FOURTH POWER OF (THETEMPERATTRE^{7}) *EXPRESSED*^{2} IN THE *ABSOLUTE*^{3} SCALE SEMICOLON

EXAMPLE II

58 Space errors (15 %) 32 Words in error (9 %)

Input Message

ATTHEL EFT END ARE THE EXTREELY SHOR T WA
UTES NT NOW N AS CO SMTC RAYS COMMA GAMM ATRAYS
COMQ AND XR AYS S E MICOOON N E XT I N OR D EW
WS INCREASING ATAVELE NGT H COME TH ESLTRA VIO
LET RAYI CO MTA THEVISIBLE LIG H T RAYS COMQ

AND 6E INFRRNED OR SO CALLEDHEAT RA YS UE MICML
ON AND FINALLY COME THE H ERTZI AN ELECTR IC
RAV ES C OMQ IN CLDING THOS E ETTSED IN R ADIO
VEERIODTHE VIS INALE RAO S OF LIG HT HAWE A D
ENGT H EX TENDING F RMM ATSOETT GREE OIN T EIG
H T TOSEITEN POINT SIX TEN MILS ION6S

Message After Initial Processing

Error-Sequences in Parentheses

AT THE LEFT END ARE (THEEXTREELY) SHORT (WAUT
ESNTNOWNASCOSMTCRAYS) COMMA GAMMA (TRAYSC
OMQAND) XRAYS (SEMICOOON) NEXT IN OR DEW (WS)
INCREASING (ATAVELENGTH) COME (THESLTRAVIOLE
TRAYICOMTATHEVISIBLE) LIGHT RAYS (COMQAND*TH*^{6}
EINFRRNEDORSO) CALLED HEAT RAYS (UEMICMLON
AND) FINALLY COME THE HERTZIAN ELECTRIC (RAV
ESCOMQINCLDING) THOSE (ETTSEDINRADIOVEERIOD
THEVISINALERAOSOFLIGHTHAWEADENGTHEXTENDING
FRMMATSOETTGREEOINTEIGHTTOSEITEN) POINT SIX
(TENMILSION*TH*^{6}S)

Results of First Letter Correction Pass

Thresholds: TYPE I 10, TYPE II 12, TYPE III 16, TYPE IV 14,
TYPE V 20

AT THE LEFT END ARE (THEEXTREELY) SHORT (WAUTES)
KNOWN^{1} AS *COSMIC*^{3} RAYS COMMA GAMMA (TRAYS)
COMMA^{1} AND XRAYS *SEMICOLON*^{5} NEXT IN OR DEW
(WS) INCREASING *WAVE*^{1} LENGTH COME THE *ULTRA
VIOLET*^{5} (TRAYICOMTATHEVISIBLE) LIGHT RAYS *COMMA*^{1}
AND THE *INFRARED*^{5} OR SO CALLED HEAT RAYS (UEMI
CMLONAND) FINALLY COME THE HERTZIAN ELECTRIC
WAVES^{2} *COMMA*^{1} (INCLDING) THOSE (ETTSEDINRADIO
VEERIODTHE) *VISIBLE*^{5} (RAOSOFLIGHTHAWEADENGTH
EXTENDINGFRMMATSOETTGREEOINTEIGHTTOSEITEM)
POINT SIX TEN *MILLIONTHS*^{4}

Results of Second Letter Correction Pass

Thresholds: TYPE I 7, TYPE II 8, TYPE III 9, TYPE IV 10,
TYPE V 11

AT THE LEFT END ARE (THEEXTREELY[7]) SHORT *WAVES*[5]
KNOWN AS COSMIC RAYS COMMA GAMMA (TRAYS[9])
COMMA AND XRAYS SEMICOLON NEXT IN *OR DEW*[11]
(WS[8]) INCREASING WAVE LENGTH COME THE ULTRA
VIOLET (*T*[9]*RAYS*[3]) *COMMA*[4] THE VISIBLE LIGHT RAYS
COMMA AND THE INFRARED OR SO CALLED HEAT RAYS
SEMICOLON[5] AND FINALLY COME THE HERTZIAN ELEC
TRIC WAVES COMMA (INCLDING[7]) THOSE (ETTSED[7]IN)
RADIO *PERIOD*[5] THE VISIBLE *RAYS*[3] OF LIGHT *HAZE*[10]
A *LENGTH*[3] EXTENDING *FROM*[4] (ATSOET[8]TGREE[14]OINT[7]EI
GHTTOSEITEN[14]) POINT SIX TEN MILLIONTHS

[1] TYPE I CORRECTION
[2] TYPE II CORRECTION
[3] TYPE III CORRECTION
[4] TYPE IV CORRECTION
[5] TYPE V CORRECTION
[6] AUTOMATIC 6 → TH CHANGE
[7] WORD CONTAINS TYPE VI ERROR
[8] WORD LACKS A CORRECT DIGRAM
[9] EXTRA T
[10] ERROR: *HAZE* PRECEDES *HAVE* ON HA DIGRAM LIST
[11] WORD GARBLED TO OTHER WORDS IN VOCABULARY
[12] AUTOMATIC 6 → TH CHANGE INCORRECT
[13] NOTE CHANGE OF INCORRECT 6 → TH SUBSTITUTION
[14] THRESHOLD TOO HIGH TO PERMIT CORRECTION

TESTS

Messages

To compile a set of test messages with a known vocabulary, sixty passages, averaging 72 words in length, were taken from three chapters of a college textbook on meterology (Blair and Fite, 1957). Hyphens and apostrophes were omitted and the remaining punctuation and the numerals were spelled out. The sixty messages contained 1061 different words of which approximately 200 were variations of another word in the vocabulary.

Ability to Determine Word Spaces

To ascertain the Degarbler's ability to determine word boundaries in a correct letter string, the sixty messages were processed by the Space routines before errors were added. (Remember the Read-in routine stores a message as a letter string and the space routines do not make

TABLE II

A Sample Portion of the MAUDE Error-Frequency Table

Correct letter(s)	Incorrect MAUDE Output	Frequency of occurrence
A	ET	2
A	T	7
ACK	WNA	1
C	GTN	1
C	K	6
C	TN	3
C	Y	1
T	E	1
TH	6	42
TI	D	3

use of the original space information). While determining approximately 4300 word spaces the program made 13 errors. In ten cases our definition of a dockable word was at fault. Example: given the letter string THEREMAINING, the program selects THERE and cannot solve MAINING. The Back-up routine fails because THERE is not dockable by our definition. In the other three cases the program found a series of words which differed from the intended one. This is a more difficult problem since the program has no way of knowing that its solution is incorrect. Example: A MOUNTAIN became AMOUNT A IN. This could be solved by very elementary context rules but choosing between PROVIDED AT A and PROVIDE DATA is more difficult. This type of problem will increase with the size of the vocabulary.

Garbled Messages

An error frequency table was compiled of all the errors which had occurred in a large sample of MAUDE output. It contains the correct letter or group of letters, the incorrect MAUDE output and the frequency of occurrence of the error. Table II contains a representative portion of the table.

Each of the sixty messages was garbled by an auxiliary program. First, 15% of the word and letter spaces were chosen at random. Each of the selected letter spaces was changed to a word space and vice versa. Then an error was picked at random from the error-frequency table, an occurrence of the correct letter(s) was located in the text and the error substituted. Errors were incorporated into the text in this manner until the number of errors equalled 9 % of the number of letters in the original text. However, the message was printed out when the error level reached 3, 5, 7, and 9 % resulting in four garbled versions of each message. The first paragraph in each of the examples shows a message after it has been garbled.

Measures of Performance

The sixty messages each garbled to contain 3, 5, 7, and 9 % error have been processed with a standard set of thresholds. The results are tabulated in Tables III and IV. Two measures were used to judge the Degarbler's performance: the gross correction percentage, i.e., the number of words corrected as a percentage of the number of words originally in error, and the relative correction or "DID/SHOULD" percentage, i.e., the number of corrected words as a percentage of the number of words the program "should" correct. A garbled word that the program "should" correct is one that contains a correct DIGRAM and a TYPE I to V error regardless of the Value of the letter string. The program occasionally fails to solve a letter string it theoretically should correct for the following reasons: (1) the Value of the letter string is below the threshold. Example: AND garbled to ANM will not be corrected if the TYPE III threshold is higher than 6; (2) A garble causes the space routines to err. Example: THE WEATHER garbled to THE REATHER. The space routines produce THERE (ATHER) (since HER is not in the vocabulary the final R is a dangling letter and (ATHER) becomes an Error-sequence.) The correction routines promptly turn this into THERE OTHER; (3) the program investigates a list of words which contains an

TABLE III

Comparative Results for 60 Messages (4341 Words), Each Processed at 4 Garble Levels

Garble level	No. of garbled words	No. of words corrected	Gross correction percentage	No. of words Degarbler should correct	Did/should percentage	No. of incorrect changes
3%	613	453	74.0%	489	92.6%	49 (8.0%)
5%	978	675	69.1%	746	90.5%	85 (8.7%)
7%	1312	874	66.6%	970	90.1%	124 (9.5%)
9%	1607	1023	63.7%	1147	89.2%	154 (9.6%)

TABLE IV

Analysis of Errors Degarbler Did Not Correct

Garble level	Number of garbled words	Number of words not corrected	Words Degarbler is designed to correct which it did not correct		Words program is not designed to correct		
			Thresh-old too high	Pro-gram[a] error	No correct digram	Word had Type VI error	Word garbled to another word
3%	613	160	16	20	70	48	6
5%	978	303	33	37	127	92	14
7%	1312	438	44	52	175	152	15
9%	1607	584	54	69	229	213	19

[a] Includes such cases as HAWE "corrected" to HAZE rather than HAVE because HAZE precedes HAVE on HA digram list, or an incorrect change made earlier in the Error-sequence which further mutilates the beginning of a garbled word so that it no longer can be corrected, etc.

incorrect trigram (or digram) and accepts one of the words as a solution. Example: THEIR garbled to THIER. The program would search the THI trigram list and accept THIS as a solution; (4) The program investigates a list of words containing a correct trigram (or digram) and accepts the first word which satisfies the correction criteria when a subsequent word on the list is the desired solution. Example: INCRE ASES garbled to INKREASES. The correction routines will produce INCREASE*D* since this word precedes INCREASES on the IN digram list.

Other errors occur when the program tries to correct a letter string which does not contain a correct digram or one which contains a TYPE VI garble for a word. Example: THE SURFACE garbled to THESE TRFAKE. The correction routines produce THESE TRUE TAKE. SURFACE was considered as a solution on the first pass because of the mutual FA digram. (On the first pass the thresholds are too high to permit the change to TRUE on the TR digram). However, since three changes would have to be made in the letter string to produce SUR FACE, it was rejected as a solution.

This type of error occurs mainly when the thresholds are lowered for the second set of correction passes. In fact, the first set of correction passes produce approximately one error for every 23.7 corrections made.

Possibly the second set of correction passes should be eliminated or the thresholds raised since they produce one error for every 2.1 corrections. False changes are undesirable as they often greatly increase the difficulty a human has in reading the message. A person with a knowledge of Morse Code might well solve the string THESETRFAKE, but after the program has changed it to THESE TRUE TAKE the meaning is probably lost.

At present the Degarbler takes between one and two minutes to process a message of, say, 70 words. In general the more errors there are in a message the longer it takes to be processed. In programming the Degarbler, no particular effort was made to reduce the program's running time.

DISCUSSION AND CONCLUSIONS

There are many improvements which could be made in the Degarbler. The correction criteria could be extended to allow the Degarbler to correct at least some of the TYPE VI words. Our present

TABLE V

DEFINITIONS

Dockable word	A word which remains a vocabulary word when its last letter is deleted. EX: THE*Y*, FORT*Y*
Subword	A word which forms a part of another word unless the two words only differ by a final S. EX: S*ITE*S
Dangling letter	A letter which the Degarbler cannot currently incorporate into a word
Error-sequence	An area of the text containing one or more dangling letters and any surrounding words which might need to be altered in order to incorporate the dangling letters into words
Mark	A signal duration in Morse Code
Dash	A mark; ideally with a relative duration of 3
Dot	A mark; ideally with a relative duration of 1
Space	The time between marks
Intraletter space	A space between marks in a letter; ideally with a relative duration of 1
Letter space	A space between letters in a word; ideally with a relative duration of 3
Word space	A space between words; ideally with a relative duration of 7
Digram	Two consecutive letters in a word
Trigram	Three consecutive letters in a word

methods fail if the garbled version of a word does not contain a correct digram (AND → PD a TYPE I error). This usually occurs in words which contain no more than four letters. We have considered storing the Morse Code pattern for such words in the computer. Then when the program succeeds in isolating in an Error-sequence the garbled version of a short word, comparing the Morse Code pattern of the Error-sequence with this table might be successful.

Undoubtedly the running time could be reduced if the redundancies of the correction routines were eliminated. For example: consider the letter string METAOFOLOGY, a TYPE VI garble for METEOROL OGY. Since there are four trigrams and six digrams in the string which are contained in the word METEOROLOGY, this word will be tried 10 times as a solution. Also any other word which contains two or more of the trigrams or digrams present in the string will be redundantly analyzed.

Although improvements could be made, it seems clear to us that the program's output is genuinely much more intelligible and useful than its input. In other words, for errors caused by noisy communications circuits, no enormous linguistic or semantic analysis is needed to provide a sensible degree of correction.

Acknowledgment

Many people contributed to the Degarble program. The original impetus came from O. G. Selfridge who with B. Gold made many valuable suggestions. F. Van Hoosen contributed substantially to the original processing rules. M. A. Wilkes wrote the auxiliary program which derives the digram and trigram lists.

Received: October 25, 1962

References

Blair, T. A., and Fite, R. C., (1957), "Weather Elements." Prentice-Hall, Englewood Cliffs, New Jersey.

Gold, B., (1959), Machine recognition of hand-sent Morse Code. *IRE Trans. on Inform. Theory,* **IT-5. 1.**