A Program for Correcting Spelling Errors

(b)(3)-P.L. 86-36

BY [redacted]

Unclassified

A description of a program using a simple, heuristic procedure for associating "similar" spellings, which is able to correct misspelled words. Given only a vocabulary of properly spelled words, the computer can correct most (including unanticipated) misspellings without human assistance. Apart from practical applications, the process is interesting as an example of an unusual form of pattern recognition.

It is tempting to assume that English spelling is too irrational to be explained to a computer. If we limit ourselves to algorithms, perhaps this is true; yet if we give the machine an extensive vocabulary, it can be programmed to recognize as misspelled any word that is not in this list. Even this procedure will not detect all errors, for some misspellings are correct spellings of different words (e.g., *advice* can become *advise*). Since such errors can only be detected through context, I avoid this troublesome prospect by considering them as usage rather than spelling errors, and so outside the scope of my title.

Having discovered a word that is not in its vocabulary, what should the program do next? Obviously, it could maintain a dictionary which associates every misspelled word with its correctly spelled equivalent. But, this auxiliary dictionary is potentially several times longer than the already sizable vocabulary of correctly spelled words. Unless the basic vocabulary is extremely limited, maintenance of the auxiliary dictionary is impracticable.

Any hope of programming customary orthographic "rules" is destroyed at first glance; for, while a machine could easily put "'i' before 'e' except after 'c' . . .", it would have difficulty recognizing " . . . and when pronounced 'a' as in neighbor and weigh". Such coding difficulties, the numerous exceptions, and the lack of rules to cover many spelling errors make this approach unpromising.

If a spelling error is correctable without reference to the context in which it appears, then the misspelling must be sufficiently "close" to the correct spelling to permit unique association. Thus, if a machine is given a suitable criterion for computing the "similarity" of words, it can "correct" a spelling error by substituting the "most similar" correctly spelled word for the misspelling. In pattern recognition terms, a misspelled word is a pattern that is approximately equivalent to its correct version. Recognizing erroneous spellings requires devising some means of dividing all spellings into equivalence classes and giving the name of the class to each of its members.

Approved for Release by NSA on 01-07-2008, FOIA Case # 51551

(b)(3)-P.L. 86-36

How is "similarity" to be measured? One immediately thinks of ad hoc rules (e.g., if all other letters are the same, a word containing "ie" is very similar to a word containing "ei");[1] but programming them introduces the same difficulties that arise in programming orthographic "rules".

One approach to associating "similar" words is exemplified by the Soundex method, which files names according to a code based on their pronunciation. To form the code, the initial letter of the surname is followed by a 3-digit number which is constructed by ignoring vowels and assigning the same digit to similar sounding consonants in their order of occurrence.[2] The filing clerk can then select the proper individual from the section of the file specified by this code on the basis of given names or other identifying information.

Although widely and successfully used by human clerks, Soundex is not readily adaptable as a machine process for correcting spelling errors. To be sure, the code construction could easily be programmed, but the fact that it associates correct spellings of different words means that an additional distinguishing criterion is required. It seemed more efficient to search for a single "similarity" measurement which normally would uniquely associate a misspelling with its correct equivalent.

An abbreviation is a particular type of "misspelling" which retains enough "similarity" to the original word to permit unique association. Unique association implies that the abbreviation retains the meaningful "kernel" of the word. A spelling error, to be recognizable without using context, must also contain the meaningful "kernel". Thus, we are led to assume that two words are "similar" if their abbreviations are identical.

An r-letter abbreviation of an n-letter word can be produced by deleting those n−r letters which are least important in the identification of the word. The problem of producing an adequate abbreviation is, in application, that of deciding which letters in a word are the least important in determining its meaning. Information theorists assume that the information conveyed by a "message" is inversely proportional to its a priori probability of occurrence. One can apply this idea by eliminating the n−r letters in the order of their expected frequency; we tried this but found that even better results can be obtained by using the "frequency" of their occurrence as errors. An empirically constructed approximation of the latter function is given in Table I. The inadequacy of this technique is soon revealed by encounters with abbreviations such as "xpnn" for exponent. Clearly weight must also be given to the position of the letter in the word. The first letter is of greatest importance, and, all other things being equal, the last letter is second in importance, followed by the second letter, the next to last letter, etc. That is, if we reorder the letters in this fashion, the desirability of rejecting a letter in a given position is an increasing, monotonic function of the new position. An empirically constructed approximation of this function is given in Table II.

Table I
The Logarithm of the Desirability of Deleting a Letter as a Function of Its Name

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

Table II
The Logarithm of the Desirability of Deleting a Letter as a Function of Its Position

Position	Score	Position	Score
1	0	9	5
2	1	10	5
3	2	11	6
4	3	12	6
5	4	13	6
6	4	14	6
7	5	15	6
8	5	16 up	7

By assuming that the name and position of a letter independently determine the desirability of rejecting it, one can form an r-letter abbreviation by deleting the n−r letters which have the largest product.[3] Although the assumption of independence is not strictly true, it is sufficiently accurate for our purposes. More refined results could be obtained by storing the larger table required for dependent variables.

Before it is asked to correct misspelled words, the machine must compute and store a short (we used 4 letters) abbreviation of each

[1] An extensive collection of such rules is given in: *Searching Aids for Alphabetic and Soundex Files*. Remington Rand Management Controls Division, New York. n.d.

[2] This statement is slightly oversimplified. For further details see: *Soundex*. .Remington Rand, New York. n.d.

[3] To minimize time and storage requirements, 3-bit logarithms are added to compute the "product." The crudity of our estimates justifies no higher precision.

(b)(3)-P.L. 86-36 [redacted]

Example

A	B	S	O	R	B	E	N	T		A	B	S	O	R	B	A	N	T
5	1	5	4	4	1	7	3	3	Letter Score	5	1	5	4	4	1	5	3	3
0	2	4	5	5	5	4	3	1	Position Score	0	2	4	5	5	5	4	3	1
5	3	9	9	9	6	11	6	4	Sum of Scores	5	3	9	9	9	6	9	6	4
		*	*	*		*	*		Delete			*	*	*		*	*	
					A	B	B	T	Abbreviation	A	B	B	T					

correctly spelled word in the vocabulary. These abbreviations are then associated with their complete spellings and sorted. The machine can now correct misspellings in any text which contains only those words in its vocabulary. Reading the words in order, it forms their abbreviations and selects all identical abbreviations of correctly spelled words. Normally this process gives a unique answer and the spelling associated with the abbreviation is then used for output (see example). When an abbreviation coincides with more than one vocabulary entry, the program compares longer abbreviations of this input word with longer abbreviations of the vocabulary entries it matched until a unique one has been selected. Of course, it is possible that a misspelling will be so extreme that its abbreviation will not appear in the vocabulary. When this happens the machine can do no more than indicate that this word was unidentifiable.

The association of common misspellings[4] with their correctly spelled equivalents is illustrated in Table III. The program correctly identified 89 of the 117 misspelled words (3 required longer abbreviations) while incorrectly identifying only 2.[5] Before condemning the machine's performance, test yourself by covering the correctly spelled column and see how well you compare. Unless you are an exceptional speller, it will be an illuminating - and humbling - experience.

The two types of deficiency are easily detectable and correctable. A word that has been incorrectly identified by the program is virtually always conspicuous because it does not fit the context and a word not identified at all is made apparent by the blank space left in the output. These errors arise either because the word was not in the original vocabulary or because the misspelling was so extreme that it gave rise to a different abbreviation. The first type of error can be corrected by simply adding the new word to the vocabulary at the next updating run. The second type requires a certain amount of "cheating". A special vocabulary updating is used in which the correct spelling of this word and the abbreviation of the particular misspelling are placed in association in the vocabulary. Although inelegant, this procedure is quite efficient in allowing for peculiar exceptions and words that are too short to permit deleting all incorrect letters while maintaining the selected length of abbreviation.

Since this heuristic process was specifically designed for the type of spelling errors normally made by people, it is considerably less effective in correcting other types of errors. It would, for example, have little utility in correcting the output of a malfunctioning machine; fortunately, however, we have other means of dealing with these. Similarly, it is not difficult to construct "misspellings" that the process will fail to correct, but it is surprisingly difficult to select such errors from the writings of people.

The author desires to acknowledge the valuable assistance of Mr. R. W. Tobin, who prepared the programs used to test these ideas.

Table III

Examples of Associating Incorrect Spellings With their Correct Equivalents by "Abbreviation"

Correct Spelling	Abbreviations	Incorrect Spelling
ABSORBENT	ABBT = ABBT	ABSORBANT
ABSORPTION	ABON ABBN	ABSORBTION
ACCOMMODATE	AMDT = AMDT	ACCOMODATE
ACQUIESCE	ACQC AQUS	AQUIESE
ANALYZE	ANYZ ANZE	ANALIZE
ANTARCTIC	ANTC = ANTC	ANTARTIC
ASININE	ASNN = ASNN	ASSININE
ASSISTANCE	ASTN = ASTN	ASSISTENCE
AUXILIARY	AUXY = AUXY	AUXILLARY
BANANA	BANA = BANA	BANANNA
BANKRUPTCY	BAKY = BAKY	BANKRUPCY
BRETHREN	BRTN = BRTN	BRETHEREN
BRITAIN	BRTN = BRTN	BRITIAN
BUOYANCY	BUYY BOYY	BOUYANCY
CATEGORY	CATY = CATY	CATAGOREY
CHAUFFEUR	CFFR = CFFR	CHAUFFUER
CHIMNEYS	CMYS CHMS	CHIMNIES
COLISEUM	COUM = COUM	COLOSIUM
COLOSSAL	COAL = COAL	COLLOSAL
COMMITMENT	COMT = COMT	COMMITTMENT
COMMITTEE	COMM = COMM	COMMITEE
CONCEDE	COND = COND	CONSEDE
CONSCIENTIOUS	CONS = CONS	CONSCIENTOUS
CONSENSUS	CONS = CONS	CONCENSUS
CONTROVERSY	COVY = COVY	CONTROVERCY
CORRUGATED	COGD = COGD	CORRIGATED
CYNICAL	CYNL SYNL	SYNICAL
DEUCE	DUCE = DUCE	DUECE
DEVELOP	DVOP = DVOP	DEVELLOPE
DIGNITARY	DGRY = DGRY	DIGNATARY
DISAPPOINT	DINT = DINT	DISAPOINT
DRASTICALLY	DRTY = DRTY	DRASTICLY
ECSTASY	ECTY = ECTY	ECSTACY
EMBARRASS	EMBS = EMBS	EMBARASS
EXAGGERATE	EXGT = EXGT	EXAGERATE
EXISTENCE	EXTN = EXTN	EXISTANCE
EXTENSION	EXTN = EXTN	EXTENTION
FEBRUARY	FBRY = FBRY	FEBUARY

[4] From: Hutchinson, L. I. *Standard Handbook for Secretaries, Seventh Edition.* McGraw-Hill, New York. 1956. pp. 133-134. Reprinted by permission.

[5] Interferred became intercede and philipinoes became Philippines. Neither of these errors would have occurred if 5-letter abbreviations had been used.

Correct Spelling	Abbreviations	Incorrect Spelling
FIERY	FIRY = FIRY	FIREY
FILIPINOS	FNOS PHNS	PHILIPINOES
FLAMMABLE	FMMB FLMB	FLAMABLE
FORTHRIGHT	FOGT = FOGT	FORTRIGHT
FORTY	FOTY = FOTY	FOURTY
FULFILL	FUFL = FUFL	FULLFIL
GNAWING	GNWG KNWG	KNAWING
GOVERNMENT	GOVT = GOVT	GOVERMENT
GRAMMAR	GRMR = GRMR	GRAMMER
HEARTRENDING	HDNG = HDNG	HEARTRENDERING
HEMORRHAGE	HMGE = HMGE	HEMORRAGE
HINDRANCE	HNDN = HNDN	HINDERENCE
HYGIENE	HYGN = HYGN	HYGEINE
IDIOSYNCRASY	IDYY = IDYY	IDIOCYNCRACY
INCENSE	INNS = INNS	INSENSE
INCIDENTALLY	INDY = INDY	INCIDENTLY
INFALLIBLE	INFB = INFB	INFALABLE
INOCULATE	INOT INNT	INNOCULATE
INSISTENCE	INTN = INTN	INSISTANCE
INTERCEDE	INTD = INTD	INTERSEDE
INTERFERED	INFD INTD	INTERFERRED
JEOPARDIZE	JODZ JPDS	JEPRODISE
KIMONO	KMNO KMNA	KIMONA
LICENSE	LINS LINC	LISENCE
LIQUEFY	LQFY = LQFY	LIQUIFY
MAINTENANCE	MANN = MANN	MAINTAINANCE
MANAGEMENT	MMNT = MMNT	MANAGMENT
MANEUVER	MAVR = MAVR	MANUVEUR
MORTGAGED	MOGD = MOGD	MORTGAUGED
NICKEL	NIKL = NIKL	NICKLE
NINETYNINTH	NNTH = NNTH	NINTYNINETH
NOWADAYS	NWDY = NWDY	NOWDAYS
OCCASIONALLY	OCNY = OCNY	OCASSIONALY
OCCURRENCE	OCNE = OCNC	OCCURENCE
PAMPHLET	PAMT PHMT	PHAMPLET
PERMISSIBLE	PRMB = PRMB	PERMISSABLE
PERSEVERANCE	PRVN = PRVN	PERSEVERENCE
PERSUADE	PRDE PURD	PURSUADE
PHILIPPINES	PHNS = PHNS	PHILLIPINES
PITTSBURGH	PBGH PTBG	PITTSBURG
PLAGIARISM	PLGM = PLGM	PLAIGARISM
PLAYWRIGHT	PWGT PLWT	PLAYWRITE
PRAIRIE	PRRE = PRRE	PRARIE
PRECEDING	PRDG = PRDG	PRECEEDING
PRECIPICE	PRPC = PRPC	PRESIPICE
PREFERABLE	PRFB = PRFB	PREFERRABLE
PRESUMPTUOUS	PRMS = PRMS	PRESUMPTOUS
PRIVILEGE	PRVG = PRVG	PRIVELEGE
PROPELLER	PROR = PROR	PROPELLOR
PSYCHOLOGICAL	PSYL = PSYL	PSYCOLOGICAL
PUBLICLY	PUBY = PUBY	PUBLICALLY
PURSUER	PURR PRUR	PERSUER
QUESTIONNAIRE	QUTR = QUTR	QUESTIONAIRE
RECIPIENT	RPNT = RPNT	RESIPIENT
RELEVANT	RVNT = RVNT	REVELENT
RENOWN	RNWN RNUN	RENOUN
REPEL	REPL RPLL	REPELL
RHAPSODY	RHDY RADY	RAPHSODY

(b)(3)-P.L. 86-36

Correct Spelling	Abbreviations	Incorrect Spelling
RHODODENDRON	RDDN = RDDN	RHODODRENDON
RHUBARB	RHBB RUBB	RUHBARB
RHYTHM	RHYM RYTM	RYTHM
SACRILEGIOUS	SAGS = SAGS	SACRELIGIOUS
SAFETY	SFTY = SFTY	SAFTY
SCISSORS	SCRS SIRS	SISSERS
SEIZE	SEZE SIZE	SIEZE
SEPARATE	SPTE = SPTE	SEPERATE
SHEPHERD	SHRD = SHRD	SHEPERD
SIMILAR	SIMR = SIMR	SIMILIAR
SINCERITY	SNTY = SNTY	SINCERETY
SOUVENIR	SOVR = SOVR	SOUVINER
SPECIMEN	SPMN SPMT	SPECIMENT
SUING	SUNG = SUNG	SUEING
SURREPTITIOUS	SUUS = SUUS	SUREPTITOUS
TRANSFERABLE	TRFB = TRFB	TRANSFERRABLE
UNPARALLELED	UNPD = UNPD	UNPARALELLED
USAGE	USGE = USGE	USEAGE
VEGETABLE	VGTB = VGTB	VEGATABLE
WEDNESDAY	WDDY = WDDY	WEDENSDAY
WEIRD	WERD WIRD	WIERD

REFERENCES

1. [redacted] "On the Recognition of Information with a Digital Computer," *Jour. ACM*, Vol. 4, No. 2 (April 1957), pp. 178-188.
2. [redacted] "Coding and Code Compression," *Jour. ACM*, Vol. 5, No. 4 (October 1958), pp. 328-330.
3. [redacted] and Friedman, E. A., "The Reconstruction of Mutilated English Texts," *Information and Control*, Vol. 1, No. 1 (September 1957), pp. 38-55.

(b)(3)-P.L.
86-36