Edgar H. Sibley
Panel Editor

An automatic spelling correcting algorithm corrects most of the 50,000 misspellings culled from 25,000,000 words of text from seven scientific and scholarly databases. It uses a similarity key to identify words in a large dictionary that are most similar to a particular misspelling, and then an error-reversal test to select from these the most plausible correction(s).

AUTOMATIC SPELLING CORRECTION IN SCIENTIFIC AND SCHOLARLY TEXT

JOSEPH J. POLLOCK and ANTONIO ZAMORA

The study of computerized correction of spelling errors has a relatively long history and remains of considerable current interest if regularly appearing papers on the topic are any gauge. Whereas early papers focused on the correction of output from optical character recognition (OCR), voice recognition, Morse code, or on spelling errors in program code, the application of most interest today is probably the correction of machine-readable text. Also, the techniques involved in spelling error correction have other important applications, for example, measuring the similarity of two strings of symbols to determine the evolutionary distance of proteins. The specialized subtopics are covered extensively in recent bibliographies by Peterson [7] and Pollock [8] and are not discussed further here.

The most suitable correction strategy for text often depends on both its nature and its source. Correcting source code for a procedural language with a small vocabulary of short words (e.g., a typical programming language or the command language for a bibliographic search system) presents quite different problems than scientific text. Similarly, OCR output contains almost exclusively substitution errors, which ordinarily account for less than 20 percent of keyboarded misspellings [9, 10].

This paper describes the correction program developed as part of SPEEDCOP (Spelling Error Detection/Correction Project), a Chemical Abstracts Service (CAS) project supported by the National Science Foundation. The program is intended not as a theoretical construct but as a useful tool for text editing. Under SPEEDCOP, approximately 25,000,000 words from seven scientific and scholarly textual databases were processed to extract over 50,000 misspellings using a dictionary equivalent to 40,000 words. This contrasts sharply with most work on spelling correction that tends to feature small dictionaries and either few or artificial misspellings. In our case, the use of real data gave credibility to the proposed solution, whereas the large dictionary revealed problems such as ambiguity that would not otherwise have been discovered. Internal reports [10–13] and papers [9, 19] describe in considerable detail how the misspellings were gathered and analyzed.

The core of the SPEEDCOP program is an algorithm for correcting only isolated misspellings that contain a single error and whose correct forms are in a dictionary. This is not as drastic a restriction as it may seem as 90–95 percent of misspellings in raw keyboarding typically contain only one error [9, 10]. The SPEEDCOP program also incorporates a common misspelling dictionary and a function word routine and is therefore not rigidly restricted to the above class of misspellings. In practice, the program corrected 85–95 percent of the misspellings for which it was designed, 75–90 percent of those whose corresponding words were in the dictionary (including multiple errors), and 65–80 percent of the misspellings overall.

THE PROBLEM

A misspelling can be corrected by two fundamentally different strategies that are referred to here as *absolute* and *relative*. In an absolute method, the correction is deduced directly from the characteristics of the misspelling and a dictionary is used only for confirmation (if at all), whereas in relative methods the correct form is selected from a dictionary.

The simplest absolute method is the historical approach, which involves constructing a dictionary of misspellings that have often occurred previously in the database and are unambiguous. For example, if the string HTE is known to occur often and only as a misspelling of THE, then it can confidently be transformed to THE whenever it is found. This approach, which is limited but very cost effective, is incorporated in the SPEEDCOP correction program.

Less extreme absolute methods are based on the letter characteristics of the misspelling. For example, a triple letter in a misspelling might be changed to a double letter and the correction confidently accepted if the new string were in the dictionary. This method can be extended to letters whose doubling and undoubling have been found to be relatively frequent causes of error. However, only a small proportion of misspellings can be corrected in this fashion. More sophisticated variants of this method—based on *n*-gram probabilities—have been used by various researchers [8].

The relative strategy involves locating words in a dictionary that are most similar to a misspelling and selecting a correction from these. Generally, the selection method is based on maximizing similarity or minimizing the string-to-string edit distance [4, 8]. Although more widely applicable than the absolute approach, the relative strategy depends on being able to define a useful measure of similarity or, better, plausibility, with respect to a model of error generation.

THE SPEEDCOP CORRECTION ALGORITHM

In SPEEDCOP we elected to generate a similarity key for each word in the dictionary and then to sort the dictionary in key order. A misspelling is then corrected by locating words whose keys collate most closely to the key of the misspelling and selecting the plausible correction(s) from these. The underlying premises are that key collation proximity is a useful measure of similarity, that sorting the dictionary is equivalent to performing all possible similarity comparisons in advance, and that plausibility is much more significant than similarity.

Similarity keys exhibit less scatter than the original strings because an arbitrary number of strings may have the same key. A key also orders the strings' features so that the collating distance between two keys is a measure of the similarity of the original strings. The key generated from a misspelling is compared with dictionary keys by a random access method; the last key to compare equal to or less than the misspelling key becomes the center of a similarity-ordered set of dictionary words that are potential corrections. In practice (see the **Error Reversal** section), the correction word was at the center of this set in over half the cases studied and very close for almost all the rest.

When the set of similar words has been retrieved, each member is tested for plausibility of correction by attempting to reverse the error operations encountered in practice. This idea was first proposed by Damerau [1] in what is probably the most useful paper in the early spelling correction literature although it doesn't seem to have been applied by later practitioners.

What is envisaged here is the correction of *isolated* misspellings. If context were to be taken into account, more elaborate strategies would be needed. Although absence of context gives rise to ambiguous corrections, this would be equally true of manual correction and one cannot reasonably expect a computer program to be superior to human beings in this respect.

THE SIMILARITY KEYS

To be effective in spelling error correction, a similarity key should retain the fundamental features of a string (word or misspelling) and be insensitive to typical spelling error operations. Many approaches to string similarity appear in the literature [4, 8].

Two obvious properties of an alphabetic string are the identity and interrelationships of the letters that comprise it. Using these properties, one can envisage a spectrum of similarity keys beginning with identity at the most rigorous end (the string itself) and continuously relaxing to only the most tenuous resemblance (perhaps an equal number of universal characters). Letter content might be expressed by a key containing the string's unique letters in a predetermined order, but this low degree of similarity would fail to distinguish many words (e.g., DEGREE and GREED) and would be adequate only for very small vocabularies. A key that incorporates letter interrelationships in a simple way would consist of the string's unique letters in their original order. This key is considerably less "relaxed" because it preserves more of the original string's information. A more rigorous key still is one that preserves all the original interrelationships. A key of this kind (proposed by Riseman et al. [5, 14, 15]) consists of a set of subscripted letter pairs A_iB_j, $i < j$, where i and j represent the positions of each of the letters in the word.

The more information the key retains, the more sensitive it is to transformations caused by misspelling and the harder it becomes to connect word and misspelling. A key must blur the identity of the string, not obliterate or replicate it. The two keys used in SPEEDCOP—the skeleton key and the omission key—were constructed so as to be compatible with the experimental evidence gathered in the data collection phase [9, 10] and to be complementary to each other.

The Skeleton Key

The similarity key originally chosen was constructed by concatenating the following features of the string (word or misspelling): the first letter, the remaining

TABLE I. Skeleton Key Examples

String	Key
CHEMOGENIC	CHMGNEOI
CHEMOMAGNETIC	CHMGNTEOAI
CHEMCAL	CHMLEA
CHEMCIAL	CHMLEIA
CHEMICAL	CHMLEIA
CHEMICIAL	CHMLEIA
CHIMICAL	CHMLIA
CHEMILUMINESCENCE	CHMLNSEIU
CHEMILUMINESCENT	CHMLNSTEIU
CHEMICALS	CHMLSEIA
CHEMICALLY	CHMLYEIA

unique consonants in order of occurrence, and the unique vowels in order of occurrence.

Some examples of string/key pairs are given in Table I. Note that CHEMICAL, CHEMCIAL and CHEMICIAL have the same key and CHIMICAL and CHIMCAL have very similar ones. Thus, CHEMICAL and these (real) misspellings sort very closely together. The rationale for this key is that (1) the first letter keyed is likely to be correct, (2) consonants carry more information than vowels, (3) the original consonant order is mostly preserved, and (4) the key is not altered by the doubling or undoubling of letters or most transpositions.

The first assumption derives from data gathered in SPEEDCOP [9, 10], which showed that only 7.8 percent of the first letters of misspellings were incorrect compared to 11.7 percent of the second and 19.2 percent of the third. Moreover, every key must possess some sort of "anchor" and the first letter seemed a reasonable choice. In terms of the second assumption, it is generally assumed in the literature [18] that vowels are less important than consonants; also, the "natural" way to abbreviate seems to be to omit vowels other than the first letter. Third, in most misspellings, unique letter order is changed only slightly. Finally, this key is insensitive to a significant number of misspelling-creating transformations such as doubling and undoubling of characters and vowel/consonant transposition. Since we regard the unique consonants in their original order as the backbone of a word, we call this key the *skeleton* key.

The skeleton key reflects the misspellings collected [9, 10] and is intuitively reasonable in that strings that "look" similar produce closely related keys. The more abstract techniques (e.g., Levenshtein distance [6, 8]) produce similarities that are counterintuitive and obscure to manual inspection and are thus unlikely to represent spelling errors.

The most vulnerable aspect of this key is its emphasis on the early consonants. The closer an incorrect consonant is to the start of a word, the greater the collating distance between the keys of the word and misspelling. In principle, this does not disable the correction technique (since one could test every word in the dictionary against the misspelling), but it may make locating the correct word impractical.

The Omission Key

Analysis of the skeleton key correction results revealed that the most frequent cause of failure to correct misspellings with parent words in the dictionary was that early consonant damage prevented the valid form from being retrieved because of the magnitude of the collating distance between the keys of the word and the misspelling. Since this occurred most often with omission errors, the omission frequency string discovered in the data-gathering phase of SPEEDCOP [9, 10] was used as the basis for a new key. Consonants were omitted from words in the following frequency order: RSTNLCHDPGMFBYWVZXQKJ. That is, R was omitted more often than any other letter; J less often. The omission key for a particular string is constructed by sorting its unique consonants in the reverse of the above frequency order and then appending the unique vowels in their original order (Table II). Note that this key is much less intuitive than the skeleton key. Strings with almost identical keys do not necessarily "look" like each other because letter content alone determines the key. Keys of this kind naturally cluster anagrams (CARAMEL, MACERAL, and CAMERAL). The omission key is much harder to construct manually than the skeleton key, but not computationally.

PLAUSIBILITY-BASED CORRECTION

Having retrieved the dictionary words most similar to the misspelling, the problem now is to find the parent or correct word. Although almost all the documented methods rely on similarity measures to do this, it seemed to us from the outset that a plausibility test based on reversing the error patterns discovered in the data-gathering phase was more realistic. A measure combining the Hamming distance of the keys used by the PLATO system [16] and the distance of the word from the center of the set method was experimented with and then abandoned when it became clear that

TABLE II. Omission Key Examples

String	Key
MICROELECTRONICS	MCLNTSRIOE
CIRCUMSTANTIAL	MCLNTSRIUA
LUMINESCENT	MCLNTSUIE
MULTINUCLEATE	MCLNTUIEA
MULTINUCLEON	MCLNTUIEO
CUMULENE	MCLNUE
LUMINANCE	MCLNUIEA
COELOMIC	MCLOEI
MOLECULE	MCLOEU
CAMERAL	MCLRAE
CARAMEL	MCLRAE
MACERAL	MCLRAE
LACRIMAL	MCLRAI

this technique was unlikely to approach the effectiveness of the error reversal technique, which finds all and only plausible corrections in the retrieval set.

Error Types

Following Gates [3], we define four basic spelling error operations: *insertion*—one extra character is inserted into the string; *omission*—one character is removed from the string; *transposition*—two adjacent characters in the string are interchanged; *substitution*—one character in the string is replaced by a different one.

These are not irreducibly primitive operations nor do they necessarily provide a unique path between word and misspelling. Rather, their usefulness lies in their correspondence to real-world error-creating operations and their ability to interconvert any pair of strings. Five error classes are defined based on these operations: Four (omission, insertion, substitution, and transposition) are created by a single application of the corresponding error operation to the correct string, whereas the fifth (multiple errors) results from the successive application of two or more (not necessarily different) error operations.

Error Type and the Intrinsic Difficulty of Correction

The number of operations required to interconvert word and misspelling varies markedly with error type. For an N-letter misspelling with a single *insertion* error, the deletion of each character in turn will regenerate the original word. If one regards substitution as the fundamental operation and deletion as substitution by a null, then correction of an insertion error requires at most N attempts. Similarly, a single *transposition* error can be reversed by inverting each pair of adjacent characters in turn for a maximum of $N - 1$ inversions or $2(N - 1)$ substitutions, a single *substitution* error can be rectified by at most $25N$ substitutions, and a single *omission* error by a maximum of $26(N + 1)$ substitutions of a letter for a null.

Clearly, there is a marked difference between insertion and transposition, which require few operations, and omission and substitution, which require many more. For practical purposes, this means that only the insertion and transposition operation can be reversed at execution time by string-distance measurement techniques that examine a large number of alternatives (see, e.g., [4, 17]).

The maximum number of operations necessary to effect an error correction increases sharply when more than one error is involved. For example, to be certain of reversing a misspelling resulting from two substitution errors, a maximum of $(625/2)(N(N + 1))$ substitutions would be required. Also, the plausibility of correction decreases sharply as the number of individual errors increases. Many words can be transformed into other perfectly valid words by even two error operations, for example, INCREASED to DECREASED. At the logical limit, any N-letter string can be transformed into any other by N substitutions and into all others by $2N^N$.

Correcting multiple errors through these error reversal techniques produces a plethora of possible corrections with low plausibilities—a poor prognosis unless the parent vocabulary is extremely small. This suggests that the error correction technique should not be too powerful or it will generate too many false "corrections".

Error Reversal

Error reversal is based on the idea that if a misspelling can be transformed into a dictionary word by reversing one of the basic error operations, then the latter word is a plausible correction. Since this technique cannot correct misspellings with more than one error, 5–9 percent of the misspellings are typically excluded a priori. However, since the technique is limited to single errors, it only needs to test words of the same length as the misspelling for substitution and transposition, words of one character more for omission, and words of one character less for insertion. This greatly reduces the number of possible comparisons. The algorithm for determining whether two strings, differing in length by 0 or 1, can be transformed into each other by a single error operation is as follows. Note how enormously simpler it is than string-distance measures.

```
1.  Find the leftmost position (P) at which the strings
    differ.
2.  IF the (potential target) word is longer than the
    misspelling AND the word from position P + 1 to
      the end is identical to the misspelling from po-
      sition P to the end
    THEN the misspelling and the word differ by an
    OMISSION error.
3.  ELSE IF the word is shorter than the misspelling
    AND the word from position P to the end is iden-
      tical to the misspelling from position P + 1 to
      the end
    THEN the misspelling and the word differ by an
    INSERTION error.
4.  ELSE IF the two strings are the same length
    THEN
          IF the two strings are identical from posi-
          tion P on
    THEN the misspelling and the word differ by a
    SUBSTITUTION error.
          ELSE IF a substring consisting of the
          (P + 1)th character of the
                  misspelling followed by the Pth
                  character of the misspelling fol-
                  lowed by that part of the misspell-
                  ing to the right of the (P + 1)th
                  character is identical to the target
                  word from the Pth character to the
                  end
          THEN the misspelling and the word differ
          by a TRANSPOSITION error.
5.  ELSE the word is not a potential correction.
```

This algorithm is applied to the retrieval set containing the dictionary words whose keys collate alphabetically closest to that of the misspelling. The center of the set is the last dictionary key to compare low or equal to the misspelling key. Because collating proximity is a measure of similarity, the probability of finding a plausible correction decreases sharply with distance from the center of the retrieval set. Virtually all the ineligible words in the dictionary have thus been excluded a priori by the collating process.

The correction search begins at the center and proceeds alternately before and after the center until either a match is found or the boundary reached if a single plausible correction is sufficient. If a single plausible correction is not sufficient, the search compares every word in the set. The length-eligible words in a 50-word span of dictionary for PLATIN are shown in Table III. The fact that only 15 of the 50 words are within one character of the length of PLATIN is a typical result. PILOT would first be checked as a potential reverse insertion error for PLATIN, then PLATING recognized as a reverse omission error. If PLATING were not present, PELITE would be examined as a potential reverse substitution error, PLUTONS as a potential reverse omission error, and so on.

In practice, it is rare for the correct word to be far from the center of the set and common for it to be the first word tested. For a typical dataset of misspellings studied, population (number of misspellings at this offset) declined with offset (absolute distance from center of set) for the skeleton key as shown in Table IV. That is, for two-thirds of the misspellings, the desired correction was at the center of the set.

TABLE IV. Skeleton Key Offset for Misspellings

Offset	Population
0	1902
1	484
2	134
3	93
4	46
5	50
6	31
7	26
8	24
9	17

THE COMMON MISSPELLING DICTIONARY

The common misspelling dictionary is considered a historical approach to correction because it assumes that misspellings that have occurred in the past will recur and can then be automatically mapped to the correct form. If misspelling creation were a deterministic process and the samples from which the dictionary was created were adequate, it would be the only method needed. However, this is not feasible because (1) most spelling errors do not recur in a reasonable text span and therefore the dictionary would consist almost entirely of useless terms, (2) 10–15 percent of misspellings are ambiguous (can be plausibly corrected to more than one dictionary word), and (3) new misspellings constantly occur and cannot by definition be corrected by a historical method. Nonetheless, within these limitations, correction via dictionary lookup remains highly efficient and very effective as it requires little computational effort and is almost 100 percent accurate.

To be included in the common misspelling dictionary, a spelling error must both occur often enough and always derive from the same word. The effectiveness of a common misspelling dictionary depends very much on the size of the sample from which it is created. A dictionary based on over 10,000,000 text words might reasonably be expected to correct 10 percent of the input misspellings [11]. Intuitively, it seems unlikely that a common misspelling dictionary could correct more than 15 percent of the misspellings even if it were generated from a very large text sample, say, 100,000,000 words.

A common misspelling dictionary is extremely small compared to both the main dictionary in a detection/correction system and the proportion of misspellings it can correct. For example, the main dictionary for SPEEDCOP contains almost 40,000 words and could usefully be enlarged, while the 256-entry common misspelling dictionary that corrected about 10 percent of the misspellings in SPEEDCOP tests is more than ample.

The small size of the common misspelling dictionary probably indicates that most misspelling is stochastic, that is, most keyboarders know how to spell most words. The frequency of a misspelling seems to be determined more by the frequency of its parent word than by the difficulty of spelling it; most errors are mechanical (typos), not the result of poor spelling. In the same way, the more frequent a letter, the more likely it is to be miskeyed [10].

OTHER CORRECTION AIDS

Two other techniques for improving the SPEEDCOP correction program were tried—suffix normalization

TABLE III. Dictionary Words Most Similar to PLATIN

	Word	Key
	PALATAL	PLTA
	PALATE	PLTAE
	PLATE	PLTAE
	PLATEAU	PLTAEU
	PELITIC	PLTCEI
	PLATED	PLTDAE
	PLEATED	PLTDEA
	PLOTTED	PLTDOE
	PELLET	PLTE
	PELITE	PLTEI
	PILOT	PLTIO
PLATIN		
	PLATING	PLTNGAI
	PLUTONS	PLTNSUO
	PLUTON	PLTNUO
	POULT	PLTOU

and a routine for concatenated function words. If a suffix variant of a dictionary word is searched, it will not be found by a simple match. If a misspelling with a certain suffix cannot be plausibly corrected, it may be useful to normalize this suffix to the one most likely to be in the dictionary and try again. Studies showed that perfect suffix normalization would on average allow an additional 3 percent of the words corresponding to misspellings to be recognized. As this represents the maximum gain from suffix normalization, the technique was judged not to be cost effective.

The function word routine is based on the observation that function words—those that fulfill a syntactic rather than a semantic role (roughly speaking, anything but nouns, adjectives, verbs, and adverbs [2])—are unusually prone to being keyed without one of their associated blanks, e.g., OFTHEIR or PRONETO. This type of error is relatively easy to rectify because of the small number of frequent function words. (One of the few traps is that an unrecognized string consisting of IN prepended to a dictionary word may be a valid negative form.) This very small routine increases corrections by only 1–2 percent, but is almost completely accurate.

AMBIGUITY RESOLUTION

Approximately 10–15 percent of the misspellings studied could plausibly be transformed into more than one valid word. ABSORBE might have been changed to ABSORB by *deletion,* to ABSORBED by *insertion,* or to ABSORBS by *substitution.*

This phenomenon has several implications for the correction strategy. First, the program should not stop searching on finding a plausible correction since this may not be the target word. Second, one may choose either to display the likeliest target or to present the user with a list of possible corrections. The first course is more appealing because, if the program's choice is correct, the user has been saved the trouble of keying the change (and possibly introducing a secondary error), whereas, if it is wrong, no harm has been done: The new misspelling is no more wrong than the original one, and the user is unlikely to be misled by the change.

Two criteria for ranking alternative corrections are the relative frequencies of the target words in the database and the probabilities of the error operations involved. In the case of ABSORBE, the relative frequency of the target words in the database is almost certainly ABSORBED > ABSORB ≫ ABSORBS, while that of the error operations is OMISSION > INSERTION ≫ SUBSTITUTION [9, 10]. Both frequency- and operation-based ambiguity decisions were investigated. Unfortunately, the frequency component of the dictionary is definitely incomplete and somewhat suspect.

Superficially one might expect the probability of a target word's being correct to follow directly the order of frequency of the error operations. Given the choice between an insertion and a transposition error, one might select the former since it is approximately twice as frequent as the latter. However, analyzing the mechanism involved leads to the opposite conclusion: Only a very small proportion of the possible insertions would give rise to a string that is also potentially a transposition error, whereas at least one of the relatively minute number of possible transpositions is known to produce an apparent insertion error. Thus, if a string could be either an insertion or a transposition error, it is more likely to be the latter. Experiments showed the precedence order to be: OMISSION = TRANSPOSITION > INSERTION > SUBSTITUTION. It seems that the error operations involved are a much more reliable guide to correct spelling than the database frequency of the possible corrections, although the latter may have some use as a subsidiary criterion when equivalued error operations are present.

THE DATA

The seven textual databases used as a source of spelling errors were: *Chemical Abstracts*; and *Biological Abstracts*; unedited text for the primary journals of the American Chemical Society; DOLE, a Chemical Abstracts Service in-house file of spelling corrections; *Chemical Industry Notes*; *Information Science Abstracts*; and *The Philosopher's Index.*

Chemical Abstracts (CA), the secondary information service issued weekly by Chemical Abstracts Service (CAS), contains titles, bibliographic data, abstracts, and author and keyword indexes for some 450,000 monographs per year—journal articles, patents, books—on essentially all aspects of chemistry and chemical engineering. The abstracts and keyword indexes were used as the source of misspellings. DOLE, an in-house file of CAS, also proved a highly suitable database as it records spelling corrections made by editors to abstracts and a range of indexes.

Biological Abstracts (BA) is a publication of abstracts analogous to CA that deals with biology rather than chemistry serials. In terms of this project, it differed most significantly from CA in that only edited text was available. In addition, BA has a reputation for containing relatively few misspellings, so one may assume that a determined effort had already been made to remove all misspellings from the text. Even so, using a sufficiently large body of text yielded a satisfactory number of misspellings.

Unedited keyboarded text for the American Chemical Society (ACS) primary journals was used as well since complete papers tend to have a rather different vocabulary than abstracts. *Chemical Industry Notes* (CIN) is a weekly ACS publication summarizing chemical industry news as reported in some 80 diverse serials, e.g., the *Wall Street Journal* and sundry trade publications. *Information Science Abstracts* (ISA) publishes abstracts of monographs in the information science field, while *The Philosopher's Index* (PI) fulfills the same role for philosophy.

These seven databases provided a reasonable diversity of vocabulary. Passing the text against a 40,000-word dictionary produced a mixture of misspellings and words missing from the dictionary. The misspellings were manually extracted and linked to their correct forms, cumulated, and classified to give records

TABLE V. Total and Unique Misspellings Collected from Each Database

Database	Words Scanned	Total Mis-spellings	Different Mis-spellings
ACS	2,937,929	5,542	4,342
ACS-2	1,787,326	2,512	2,050
BA	4,645,593	4,662	3,905
CA	4,762,128	10,243	3,026
CIN	756,835	1,718	1,491
DOLE	10-15,000,000	27,844	21,335
ISA	118,950	362	335
PI	38,568	80	72
Total	25-30,000,000	52,963	31,815

TABLE VI. Correction Ceiling for Each Database

Dataset	Dictionary Loss (%)	Multiple Errors (%)	Ceiling (%)
ACS	7.49	5.90	86.61
ACS-2	7.93	6.34	85.73
BA	23.90	4.85	71.25
CA	5.12	3.48	91.40
CIN	22.03	4.43	73.54
DOLE	12.27	8.97	78.76
ISA	20.31	4.86	74.83
PI	34.72	6.95	58.33

containing the misspelling, the corresponding word, a code identifying the error type, and the frequency of the particular word/misspelling pair. (See [9, 10] for details.) In the following discussion, a dataset of misspellings is referred to by the acronym of the database from which it derives.

The correction program was developed using over 50,000 misspellings extracted from some 25,000,000 words of text from the above databases (Table V). Note that because of the nature of DOLE, the number of words scanned can only be estimated and therefore the upper limit is more likely to be correct. Also, the Different Misspellings column does not sum in a simple way since there are duplicate misspellings between datasets as well as in them.

The correction program was then retested on a fresh set of misspellings (ACS-2) extracted from the ACS database. Both the composition of ACS-2 and its correction results were completely consistent with those of the other datasets.

EXPERIMENTAL RESULTS

The correction program consists of the following modules, each of which is applied only if the previous one(s) fails to yield a correction: the common-misspelling dictionary, the skeleton key, the omission key, and the function word routine. Each dataset of misspellings was run both with and without the common misspelling dictionary because, although this would certainly be used in an operational system, the most interesting part of the program is undoubtedly the similarity key/error reversal correction algorithm.

The Scope of the SPEEDCOP Algorithm

There are three ceilings against which one might measure the success of the correction algorithm. First, since the method is restricted to single-error misspellings whose correct forms are in the dictionary, one might take the percentage of these errors corrected as a measure of success and ignore the remaining misspellings. Or, one might argue that a more powerful algorithm would correct any misspelling whose valid form is in the dictionary. Finally, one might expect a correction program to correct all misspellings. (These three measures are labelled, respectively, ALGORITHM, RELATIVE, and ABSOLUTE in Table VII.) The ABSOLUTE measure is difficult to sustain because, unlike the limitations inherent in the ALGORITHM and RELATIVE measures, the adequacy of the dictionary is independent of the correction algorithm. The effectiveness of the correction program is certainly affected by the dictionary composition, but that of the algorithm is not. The theoretical correction ceiling for each dataset is shown in Table VI, where Dictionary Loss refers to the percentage of target words not in the dictionary and Ceiling to the best result achievable by an algorithm limited to single errors with correct forms in the dictionary.

The dictionary essentially reflects the vocabulary of the CA titles and keyword index, so it is naturally less than ideal for the other databases. Table VI shows how

TABLE VII. Effectiveness of the Correction Program as Measured by Three Different Criteria

Dataset	Algorithm (%)	Relative (%)	Absolute (%)
ACS	82	77	71
ACS-2	88	83	74
BA	94	88	67
CA	84	81	77
CIN	84	79	62
DOLE	83	74	65
ISA	77	77	57
PI	96	86	56

TABLE VIII. Cumulative Effectiveness of the Similarity Keys

Dataset	Corrected (%)		Miscorrected (%)		Uncorrected (%)	
	S-Key	O-Key	S-Key	O-Key	S-Key	O-Key
ACS	65.81	71.00	10.85	11.54	23.34	17.45
ACS-2	68.51	74.40	9.20	9.75	22.29	15.84
BA	61.04	66.87	5.25	5.89	33.71	27.24
CA	71.73	76.95	12.36	12.76	15.91	10.29
CIN	57.61	61.89	11.42	12.18	30.97	25.94
DOLE	58.73	65.16	11.10	11.81	30.17	23.04
ISA	54.02	57.34	11.91	12.47	34.07	30.19
PI	48.75	56.25	3.75	3.75	47.50	40.00

TABLE IX. Correction of Error Types for ACS Misspellings

Key	Operation	Corrected (%)	Miscorrected (%)	Uncorrected (%)
SKELETON	OMISSION	67.16	8.03	24.81
	INSERTION	78.94	9.76	11.30
	SUBSTITUTION	53.39	17.99	28.62
	TRANSPOSITION	79.33	7.91	12.76
	MULTIPLE	0.00	20.99	79.01
	TOTAL	65.81	10.85	23.34
OMISSION	OMISSION	20.24	4.17	75.60
	INSERTION	40.68	2.26	57.06
	SUBSTITUTION	19.59	3.67	76.73
	TRANSPOSITION	66.30	0.00	33.70
	MULTIPLE	0.00	1.57	98.43
	TOTAL	5.19	0.70	17.45
BOTH	OMISSION	72.30	9.09	18.61
	INSERTION	83.54	10.02	6.44
	SUBSTITUTION	59.81	18.23	21.96
	TRANSPOSITION	86.96	8.74	4.30
	MULTIPLE	0.00	22.22	77.78
	TOTAL	71.00	11.54	17.45

important an adequate dictionary is to the correction algorithm.

The Similarity Keys

Table VII shows the correction results achieved using only the similarity keys (the skeleton and omission keys), whereas Table VIII gives their cumulative effectiveness. The skeleton key routine was used first, but the final result was found to be essentially independent of the order in which the keys were invoked. The results in Table VIII are cumulative and refer to the total input; that is, 65.81 percent of the ACS misspellings were corrected by the skeleton key and when those uncorrected were passed to the omission key routine, the omission key corrected a number corresponding to 5.19 percent of the total input to give a cumulative result of 71.00 percent. Thus, the second similarity key in effect corrects an additional 5–7 percent of the misspellings and miscorrects (transforms to an inappropriate dictionary word) another 0.5–0.7 percent.

Error Type and Correction

The relationship between error type and correction is exemplified by the results for the ACS dataset, which contains 4,342 different and 5,542 total misspellings. Without the common misspelling dictionary, the results were as shown in Table IX. The percentages in the columns refer to the specified operations and the TOTAL percentages to all the input. That is, for OMISSION errors the skeleton key corrected 67.16 percent, selected the wrong word for 8.03 percent, and was unable to find a plausible correction for the remaining 24.81 percent. With regard to the whole input dataset, the corresponding figures are 65.81 percent, 10.85 percent, and 23.34 percent. Similarly, of the 23.34 percent passed unchanged by the skeleton key routine, the omission key routine corrected a number equivalent to 5.19 percent of the total input, assigned the wrong target word to 0.70 percent, and ignored 17.45 percent. The OMISSION key TOTAL thus represents the absolute increase in CORRECTED and MISCORRECTED, whereas BOTH shows the final results of applying the keys sequentially to uncorrected misspellings. (The results at each stage are essentially independent of the order of application of the similarity keys.)

Substitution errors are treated the least successfully but do not account for a disproportionate number of miscorrections, as shown in Table X where TOTAL is the percentage contribution of each type to the total number of miscorrections and N-TOTAL the normalized contribution of each type to the total percentage of miscorrections (i.e., (TOTAL x 100)/9.75 for this dataset). Thus, although over 18 percent of the substitution errors were miscorrected, this constituted only 2.87 percent of total misspellings because of the relatively small proportion of substitution errors, but over 29 percent of the miscorrections, which in turn were generated by only 9.75 percent of the misspellings input. It is interesting that 19 percent of the multiple errors lay within one error operation of a dictionary word.

The Common Misspelling Dictionary

Unlike the correction algorithm, the common misspell-

TABLE X. Effect of Error Type on Miscorrection Rate

Operation	Miscorrected (%)	Total (%)	N-Total (%)
Omission	7.03	2.71	27.8
Insertion	10.48	2.59	26.6
Substitution	18.41	2.87	29.4
Transposition	2.68	0.40	4.1
Multiple	18.63	1.19	12.3

TABLE XI. Effectiveness of the Common Misspelling Dictionary

Dataset	Corrected (%)	Miscorrected (%)	Uncorrected (%)
ACS	12.09	0.46	87.45
ACS-2	9.43	0.36	90.21
BA	6.60	0.09	93.31
CA	28.55	0.34	71.11
CIN	10.77	0.35	88.88
DOLE	8.67	0.26	91.07
ISA	12.47	0.28	87.26
PI	2.50	0.00	97.50

TABLE XII. Overall Correction Results

Dataset	Corrected (%)	Miscorrected (%)	Uncorrected (%)
ACS	76.05	8.66	15.29
ACS-2	77.71	7.92	14.37
BA	68.68	5.58	25.74
CA	85.21	5.59	9.19
CIN	65.11	10.48	24.41
DOLE	69.11	9.42	21.46
ISA	62.60	9.70	27.70
PI	56.25	3.75	40.00

ing dictionary is capable of correcting multiple errors, but its usefulness in practice is limited to unambiguous and frequent misspellings (e.g., ANND → AND). The common misspelling dictionary used in SPEEDCOP contains the 256 most frequent misspellings in the merged datasets. Table XI shows its effectiveness for each dataset.

Only about 2 percent of the corrections attempted by the common misspelling dictionary failed; also, the computational effort required by this technique is trivial. However, its effectiveness as a method is limited and the dictionary needs to be based on a large sample of text (say 50–100 million words) to reach its full potential.

The effect of the common misspelling dictionary is to transfer entries from the MISCORRECTED to the CORRECTED column, not to reduce the number of UNCORRECTED misspellings, perhaps because the dictionary tends to contain short misspellings. (See [11] for a detailed discussion of the construction, composition, and effectiveness of common misspelling dictionaries.)

Overall Results

In practice, one would employ both the common misspelling dictionary and the function word routine (see under "Other Correction Aids"), which later added 1–2 percent to the correction result. Using both the common misspelling dictionary and the function word routine, the correction results were as shown in Table XII.

Effect on Valid Words

Misspellings were collected in SPEEDCOP by passing the text through a spelling error detection program and editing the output manually. When the valid words in the output were also submitted to the correction program to simulate a completely automatic detection/correction system, 26 percent were changed because they lay within one error operation of a dictionary word, for example, were suffix variants. Note that this is not a defect of the correction algorithm, but rather a manifestation of the inadequacy of the dictionary. A substantial proportion of the transformed valid words have less than five characters, a predictable correspondence with ambiguity.

Table XIII shows the effect that perfect suffix (SUFFIX) and suffix plus prefix (AFFIX) normalization [12, 13] would have on the lookup of the valid words corresponding to the misspellings. This represents the maximum possible improvement since the normalization would not apply to misspellings that affected affixes.

Ambiguity

The major problem of the correction algorithm is ambiguity. For example, using the common misspelling dictionary with the ACS-2 dataset resulted in the conversion of 7.92 percent of the ACS-2 misspellings to the wrong dictionary word; without the misspelling dictionary, 9.75 percent were converted to the wrong word. There are several explanations for these errors. First, the correct form of the misspelled word may not be in the dictionary, although the misspelling may lie within one error operation of a dictionary word. Alternatively, the correct word may be in the dictionary, but not identifiable by the algorithm (usually true only for multiple errors), whereas another dictionary entry is a plausible correction. Nonetheless, the most frequent cause of miscorrection is failure of the ambiguity resolution procedure, an inevitable problem when a large dictionary is involved.

The problem is analogous to asking a person to correct a list of isolated misspellings. Without context, a human editor would probably be no more successful than the correction algorithm (Table XIV), the chief difference being that a person would generate fewer alternatives. Given context, however, a person would resolve the ambiguities easily and accurately. Unfortunately, the same is not true of an algorithmic approach. As most of the necessary contextual information is semantic and pragmatic rather than syntactic, an extremely elaborate algorithm would be needed. The one advantage a program might have over a human editor

TABLE XIII. Maximum Improvement Possible via Affix and Suffix Normalization

Dataset	Suffix	Affix
CA	1.44	1.67
ACS	2.82	3.71
DOLE	3.28	—
BA	4.86	5.99
CIN	6.05	6.39
ISA	7.50	7.50

TABLE XIV. Plausible Miscorrections

Misspelling	Word	Miscorrection	Frequency
CHROMOPHORS	CHROMOPHORES	CHROMOPHORE	1
COMPARIBLE	COMPARABLE	COMPATIBLE	1
CONSTANS	CONSTANT	CONSTANTS	1
ONSTANT	CONSTANT	INSTANT	1
CONTANTS	CONSTANTS	CONTENTS	2
CONSISTUENT	CONSTITUENT	CONSISTENT	1
CONTAING	CONTAINING	CONTAIN	1
CONTINED	CONTINUED	CONTAINED	1
COVERGED	CONVERGED	COVERED	1
CORRESPONDSD	CORRESPONDED	CORRESPONDS	1
CHOLD	COULD	CHILD	1
ODATE	DATE	IODATE	1

TABLE XV. Variation of Average and Maximum Ambiguity with Length

Length	Popn. (%)	Ambig. (%)	Av. Ambig.	Max. Ambig.	Correct (%)	Incorrect (%)
3	2.95	81.08	4.10	10	17.57	77.03
4	6.29	47.47	3.95	10	52.53	41.41
5	7.05	22.03	2.46	4	71.75	14.12
6	8.36	19.05	2.42	5	79.05	12.38
7	10.99	15.58	2.35	4	76.81	7.61
8	12.42	9.62	2.23	4	79.81	7.37
9	14.33	4.72	2.24	4	80.28	2.22
10	13.30	4.79	2.00	2	82.63	2.69
11	9.20	3.03	2.00	2	76.19	1.73
12	7.21	4.42	2.00	2	80.11	2.21
13	4.74	0.84	2.00	2	70.59	0.84
14	1.83	0.00	0.00	0	69.57	0.00
15	1.35	0.00	0.00	0	50.00	0.00

would be incorporation of database-specific word-frequency data.

There is a strong correlation between the length of a misspelling and both the probability of its being miscorrected and the number of plausible corrections it generates. Table XV shows that 3- and 4-character misspellings are much more likely to be ambiguous than longer ones and therefore to be miscorrected. (AV. AMBIG. denotes the average number of choices for misspellings for which there is more than one plausible correction.) In fact, although 3–4 character misspellings constitute only 9.24 percent of total misspellings, they generate 42 percent of the miscorrections. Table XVI shows to what degree a multiplicity of choices lowers the correction rate. However, it is important to note that over 86 percent of the misspellings are unambiguous. The absolute improvement due to perfect ambiguity resolution (when the common misspelling dictionary is used) for the various datasets would be, in percentages: ACS, 5.04; ACS-2, 4.28; BA, 1.75; CA, 3.30; CIN, 5.33; DOLE, 5.05; ISA, 6.38; PI, 2.61.

CONCLUSIONS

The similarity-key/reverse-error correction algorithm is very effective in locating words most similar to misspellings in a large dictionary and has little scope for improvement in this respect. It requires much less computation than string distance measures and produces only plausible corrections. Although the algorithm is limited to misspellings with a single error, these single-error misspellings typically represent over 90 percent of real-world keying errors. In practice, however, correction results are crucially dependent on the main dictionary. If the dictionary is very comprehensive, ambiguity increases; if it is too small, valid words may be

TABLE XVI. Variation of Accuracy with Number of Choices

Choices	Misspellings (%)	Corrected (%)	Miscorrected (%)
0	15.80	0.00	0.00
1	70.90	94.83	5.17
2	7.70	68.56	31.44
3	2.50	43.75	56.22
4	1.20	41.38	58.62
5	0.67	29.41	70.59
6	0.72	5.56	94.44
7	0.16	0.00	100.00
8	0.24	16.67	83.33
9	0.00	0.00	0.00
10	0.16	75.00	25.00

converted to semantically unrelated forms (RAT → RAG) or to suffix variants (RAT → RATS).

A small common misspelling dictionary is a highly cost-effective and virtually error-free method of correcting 10–15 percent of the total misspellings. In contrast, even a perfect suffix normalization routine would be only a minor enhancement to a correction program with access to a reasonably comprehensive dictionary.

Most of the problems of automatic correction result from short (3- and 4-character) misspellings, and excluding these would greatly reduce the practical difficulties. Similarly, misspellings with more than one error are almost certainly intractable, thus excluding 5–10 percent of misspellings from the correction process.

The results for the different databases are encouragingly similar, suggesting that the correction program may be applicable to virtually any textual database. However, SPEEDCOP is designed for large volumes of keyed English text. Other input methods (OCR, voice, Morse code) or data types (e.g., programming languages) pose somewhat different problems.

Acknowledgment. We gratefully acknowledge the support provided by the National Science Foundation under Grant IST-7821075.

Copies of the SPEEDCOP project reports may be obtained by writing to:

SPEEDCOP
Marketing Communications
Chemical Abstracts Service
Box 3012
Columbus, OH 43210

REFERENCES

1. Damerau, F.J. A technique for computer detection and correction of spelling errors. *Commun. ACM 7,* 3 (Mar. 1964), 171–176.
2. Fries, C.C. *The Structure of English.* Harcourt Brace, New York, 1952.
3. Gates, A.I. *Spelling Difficulties in 3867 Words.* Bureau of Publications, Teachers College, Columbia University, New York, 1937.
4. Hall, P.A.V., and Dowling, G.R. Approximate string matching. *Comput. Surv. 12,* 4 (Apr. 1980), 381–402.
5. Hanson, A.R., Riseman, E.M., and Fisher E. Context in word recognition. *Pattern Recogn. 8,* 1 (Jan. 1976), 35–45.
6. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys.-Dokl. 10,* 8 (Feb. 1966), 707–710.
7. Peterson, J.L. Computer programs for detecting and correcting spelling errors. *Commun. ACM 23,* 12 (Dec. 1980), 676–687.
8. Pollock, J.J. Spelling error detection and correction by computer: Some notes and a bibliography. *J. Doc. 38,* 4 (Dec. 1982), 282–291.
9. Pollock, J.J., and Zamora, A. Collection and characterization of spelling errors in scientific and scholarly text. *J. Am. Soc. Inf. Sci. 34,* 1 (Jan. 1983), 51–58.
10. Pollock, J.J. SPEEDCOP: Task Al—quantification. *CAS Internal Rep.*, Chemical Abstracts Service, Columbus, Ohio, July 1980.
11. Pollock, J.J. SPEEDCOP: Task B1—automatic correction of common misspellings. *CAS Internal Rep.*, Chemical Abstracts Service, Columbus, Ohio, Oct. 1981.
12. Pollock, J.J. SPEEDCOP: Task B2—automatic correction of misspellings. *CAS Internal Rep.*, Chemical Abstracts Service, Columbus, Ohio, May 1981.
13. Pollock, J.J. SPEEDCOP: Task C.—evaluation of spelling error detection/correction system, *CAS Internal Rep.*, Chemical Abstracts Service, Columbus, Ohio, Sept. 1981.
14. Riseman, E.M., and Ehrich, R.W. Contextual word recognition using binary digrams. *IEEE Trans. Comput. C-20,* 4 (Apr. 1971), 397–403.
15. Riseman, E.M., and Hanson, A.R. A contextual postprocessing system for error correction using binary *n*-grams. *IEEE Trans. Comput. C-23,* 5 (May 1974), 480–493.
16. Tenczar, P., and Golden, W. Spelling, word, and concept recognition. *Computer-based Education Research Laboratory Rep. X-35,* Dept. Univ. Illinois, Urbana, Il. 1972.
17. Ullmann, J.R. A binary *n*-gram technique for automatic correction of substitution, deletion, insertion, and reversal errors in words. *Comput. J. 20,* 2 (Feb. 1977), 141–147.
18. Yum, K.S. An experimental test of the law of assimilation. *J. Exp. Psychol. 14* (Feb. 1931), 73–74.
19. Zamora, E.M., Pollock, J.J., and Zamora, A. The use of trigram analysis for spelling error detection. *Inf. Process. Manage. 17,* 6 (Jan. 1983), 305–316.

CR Categories and Subject Descriptors: H.4.1 [**Information Systems Applications**]: Office Automation—*word processing*; I.7.1 [**Text Processing**]: Text Editing—*spelling*
General Terms: Algorithms
Additional Key Words and Phrases: spelling correction, similarity keys, dictionary lookup

Received 2/83; revised 7/83; accepted 9/83

Authors' Present Addresses: J.J. Pollock, Chemical Abstracts Service, P.O. Box 3012, Columbus, Ohio 43210. A. Zamora, IBM Corporation, 18100 Frederick Pike, Gaithersburg, Md. 20879.