A Spelling Program for Use with Optical Scanners

E. Michelle Rhone

A Major Report

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

April 1987

ISBN 0-315-35509-3

ABSTRACT

A Spelling Program for Use with Optical Scanners

E. Michelle Rhone

Spelling programs are an important tool for word processing and document preparation. In this paper, a system for spelling checking and correction designed for use with optical scanners is described. The system checks for errors which are due to scanning errors and uses a probability based algorithm to select a correction without user intervention. The aim of this project is to try to determine if the program implemented has any advantages over other commercial checkers when they are used on text produced by optical scanners.

The performance of the spelling program is evaluated in two ways:

1. The group of documents analysed to form the program's probabilistic heuristics is, in turn, checked and corrected by the spelling program.

2. The spelling program is compared to three other commercial programs.

The results of these tests are mixed. Special checking and correcting heuristics are helpful for a good performance of a spelling program but perhaps the best correction method of misspelled words is manual (user corrected).

ACKNOWLEDGEMENTS

I thank my supervisor, Dr. C. Y. Suen, for his help in the planning and writing of my project and report.

I also thank Michael Yu for his help in this use of his programs and the optical scanner equipment.

Finally, I thank the Natural Sciences and Engineering Research Council of Canada for partial support of this project.

TABLE OF CONTENTS

1. INTRODUCTION

Spelling checkers and correctors have become an important tool in document preparation by word processing systems within the last several years. These spelling packages have been designed for use on document texts residing in computer files, which may contain spelling mistakes made by the author(s) or typist(s) of the documents. Spelling checkers try to point out the spelling mistakes in documents. Some spelling correctors which make an attempt to correct spelling errors have been developed.

New technology has added another way in which to easily enter a document into a computer file -- scanning the image of the document with an optical scanner. The scanner reads the page and creates a text file of identified characters. However, the software is not perfect as it can make a number of mistakes as it scans.

Spelling checkers which are designed to detect human errors may not perform as well on spelling errors made by optical scanners. In this paper, a system for spelling checking and correction designed for use with optical scanners is described. The system checks for errors which are due to scanning errors and uses a probability based algorithm to select a correction without user intervention. This spelling program is also evaluated in various ways. The aim of this project is to try to determine if the program implemented has any advantages over other commercial checkers when they are used on text produced by optical scanners.

2. GENERAL SPELLING SYSTEMS

Most spelling errors are generated in the following ways [ref. 1]: transposition of adjoining letters, insertion of an extra letter, deletion of a letter of the word, substitution of one letter of the word by another letter, and any combination of these. These errors are likely to be the result of keyboard errors (i.e. striking the wrong letter on the keyboard during input) or ignorance errors (i.e. user not knowing the correct spelling of a word). A word is usually identified as containing an error when it is not found in the spelling program's dictionary. Candidate corrections are formed by subjecting a misspelling to these transformations (usually it is assumed that only one transformation occurs per misspelling) and then searching for the newly formed words in the dictionary. All words subsequently found in the dictionary are considered as possible corrections to the misspelled word.

The basic unit for spelling programs is the word. A word is usually defined as at least two or three alphabetic characters delimited by certain nonalphabetic characters like blanks and hyphens.

Another major part of the system is the dictionary. The dictionary contains a representation of all words recognized by the system. It is usually an alphabetized sequential list which has been compressed in some way. This list could have various representations. It could be some sort of tree structure, or a partial or complete hash table. It can be compressed as well, using codes, numbers, or bit combinations to signify the appropriate letter combinations in the word. This list of words can be an exact representation of all the words in the dictionary (a one-to-one mapping of code word to dictionary word) or an approximate one (a one-to-many mapping of code word to dictionary word). The advantage of an exact list is one of accuracy while the advantage of an approximate list is one of space. Compression schemes have been extensively researched and some references can be found in the bibliography. ([2],[3],[8],[9],[10])

Many spelling packages provide one or more auxiliary dictionaries in addition to the standard one. The content of these dictionaries is user dependent and allows the user to tailor the system to the particular project. Special words not found in the standard dictionary but often used by the particular application can be stored in this user dictionary or dictionaries. When the word search is conducted, the system can be instructed as to which special dictionaries should be checked in addition to the standard one.

The spelling checker/correctors basically work in the following manner. Each word is retrieved from the document file. Sometimes, affix normalization is applied to the retrieved word. After the word form is finalized, the dictionary is searched for an occurrence of the word.

A spelling package defines a misspelled word as a word not found in one of its dictionaries. Packages can differ in the words they actually count as misspelled, however, because of differences in dictionary contents, differences in the definition of a document word, in extent and method of affix analysis, use of supplemental dictionaries, and case sensitivity, i.e. whether the distinction between upper and lower cases is preserved. Distinguishing upper and lower case is difficult (how do you handle capitalized words at the beginning of sentences?) and most commercial packages simply ignore case differences.

If a word is found in the dictionary, it is assumed to be correctly spelled, although this is not always true. A misspelled word (one not found in the dictionary) is indicated to the user in some way, usually either by highlighting it (in an interactive system) or by putting it in a list of misspelled words (in a batch system). At this point, the job of the spelling checker is done and the user must decide whether or not to correct the misspelled words. In the case of a spelling corrector, if the word is misspelled, an attempt is made to furnish the user with a list of possible corrections from which the user can choose or to correct it without user intervention, if the user wishes it.

An optical scanner converts the characters of the text being scanned to images composed of dots or pixels. For each letter, it analyses these pixels and compares the results to the characteristics of model characters available to the scanner program. On the basis of this comparison, an identifying label or character for the image is chosen. Most of the time, this character is the correct one but mistakes do occur (the numeral one (1), is often mistaken for the lower case 'el' (l), for instance). Consequently, a new method of producing spelling errors in a document file now exists. Scanners generate errors from only three of the four above ways: deletion, insertion, and substitution. Transposition errors do not occur. Also, any spelling errors in the original document will probably exist in the scanned document as well.

Normal spelling checkers dismiss numerals and punctuation marks when they retrieve a word from a file. Words which have been misspelled by humans usually differ from their correct spellings by a combination of letters. Therefore, even misspelled words will be correctly retrieved from a document using the usual definition of a word. However, words which have spelling errors produced by optical scanners could contain numerals and punctuation marks as part of a scanned word. Then, if the usual definition of a word is used, a scanned word containing a spelling error has more of a chance of being retrieved wrongly. For instance, consider the word 'nod'. An optical scanner could produce the word 'n0d' from 'nod', mistaking the zero (0) for the letter oh (o). Since commercial spelling checkers ignore numerals, they will read the word 'n0d' as 'n' and 'd'. In addition, since these spelling checkers do not deal with words of one letter, 'n' and 'd' are ignored as well. Thus, this error could go completely undetected. This situation could make it hard to identify just where spelling errors occur and would make it next to impossible to correct them. Thus, the nature of a spelling error produced by an optical scanner could be very different. This difference may not be detected by a 'normal' spelling checker.

The program written for this project implements a spelling checker which is designed to check document files produced by an optical scanner. The design and manipulation of the dictionary, a major part of a spelling checker, is not greatly affected by the type of error in the text. However, the definition of a document word and its subsequent manipulation and possible correction if misspelled will be greatly affected. This package offers the opportunity to check for and correct a scanned document file in a batch mode.

3. WORD DEFINITION

A word is defined as a sequence of one or more of the following characters: all upper and lower case letters, all numbers, and all punctuation marks except delimiters. A word is delimited by spaces, dashes, and underlines. The characters close quote/apostrophe (’), open quote (‘), double-quote ("), and comma (;) are stripped from both ends of the word if present. The characters exclamation point(!), right parenthesis()), period (.), colon (:), semi-colon (;), and question mark (?) are stripped from the tail end of the word if they are present. The characters dash (-) and underline(_) are stripped from the front of the word if they are present. These rules are slightly different from the usual ones employed by the commercial spelling checkers. Commercial programs do not have to worry about numerals and punctuation marks making up the misspelled words as well as letters (except in rare cases of certain keyboard entry errors). Consequently, they do not make allowances for such problems in their checking process. The problem of numerals and punctuation marks frequently occurring within scanned words means that a spelling program tailored for an optical scanner must use slightly different rules in the checking process. The complicated rules governing the presence of punctuation marks and numerals in scanned words are needed in order to correctly determine which words are misspelled. Since many letters are often mistaken as non-alphabetic characters by the scanner, as many of them as possible should be allowed to make up a word along with the letters. However, care must be taken that actual punctuation marks are not mistakenly included in a word. For this reason, some punctuation characters are not allowed to appear in words, and some of them must be taken off the ends of the words.

The alternative to stripping selected punctuation marks from the ends of words is to tag as incorrect any word occurring next to any punctuation mark, even if they are correct. These words will then be counted as insertion errors and can be corrected in the correction process. This may slow the program down, as most of these words will be correct yet all will be tagged as incorrect and will have to be corrected. Also, if such a word happens to already be incorrect, the addition of the punctuation mistake would make it impossible to later correct the word. This is because words are assumed to contain only one spelling error and hence are corrected for only one error. This could adversely affect correction rates. Consider the example of having an '!' mistaken for an el (l) at the end of a word. In this case, the '!' would be stripped from the end of the word and information about a possible correction is lost because it is known to the correction algorithm that '!' is often mistaken for 'l'. However, if the '!' is retained, all words which end exclamatory sentences will be found to be wrong (e.g., Oh boy! - boy! is a misspelling). This would affect the program speed and could mask any errors already existing in the word. For this reason, the former process of stripping punctuation marks where appropriate is used by this spelling package.

Finally, words more than the maximum length are checked for and broken into parts if they occur. There is no way at present that most words which are too long can be corrected. The maximum word length allowed in this package is 24 letters. This restriction does not have any adverse affect on the checking method because short words have a greater frequency of occurrence than long ones. Words with length greater than than 24 will have an occurrence frequency close to zero.

4. THE DICTIONARY

The dictionary is a simple sequential list which is alphabetized according to the ASCII collating sequence. This means that all upper case letters are lower in sequence than all lower case letters, and the former appear in the list before the latter. For example, 'Peter', 'Canada', 'cat', and 'an' would be ordered as 'Canada', 'Peter', 'an', and 'cat'. There is no special reason why this particular sorting scheme is used. It is simply the one provided with the language tools which were used to generate the spelling program. The dictionary is contained on a disc file.

In order to be effective, the dictionary must contain thousands of words. It is important that the dictionary be large enough to cover words most likely to be encountered in any random text. On the other hand, a dictionary should not contain many rare or obscure words as they are more likely to match other misspelled words than they are to occur in a document. A large dictionary will take up a lot of storage, however there are many ways such a dictionary can be compressed. The dictionary used in this package employs a simple compression scheme which provides an exact representation of all dictionary words and has no loss of information. A word is compressed based on the word proceeding it. First, the word is analysed to determine if it has a set of beginning letters identical to a set of beginning letters of its predecessor word. The word to be compressed is then coded as the length of the longest such set (which could be zero) and the rest of its letters. Thus the list 'wall', 'walk', 'walking' is compressed as 'wall', '3k', '4ing'. Since 'wall' is the first word in the list, none of its letters can match anything and it is coded as 'wall'. However, in the word 'walk', the first three letters are the same as the first three letters of the preceding word 'wall', so 'walk' is coded as the length of the set (3) plus the rest of the letters in the word (k) or '3k'. For 'walking', the first 4 letters are the same as in 'walk' which is its predecessor and it is coded as '4ing'. In order to take full advantage of this scheme, the dictionary is alphabetized and sorted in the way previously described, in the ASCII collating sequence. The savings for a small dictionary are usually not much but as the dictionary grows, the word combinations get more redundant and compression will save a lot of space. The dictionary used by this spelling package has approximately 60,000 words and is compressed about 50 per cent from the original list. The advantages of the compression scheme are that the dictionary is easy to read as an ASCII file, the dictionary is easy to form and to search, and there is no loss of information. With an index, the searches are relatively fast as well.

The index for the dictionary is contained within the actual program. It is a 26 member array organized so that each array member points to the starting byte of an alphabetized section signified by a new beginning letter.

5. CHECKING METHOD

After the scanned word is retrieved from the document file, some spelling programs manipulate and analyze it before checking whether it is contained in the dictionary. Two such processes are affix normalization and case analysis. Affix normalization occurs when selected suffixes and prefixes are removed from the word, reducing it to its stem. All words are made up of a stem, prefixes (e.g. anti,multi,un,sub), and suffixes (e.g. s,ed,able,ing). An affix is either a prefix or a suffix. A word may have any number of affixes, including none. There is no fixed list of affixes and in some cases an affix in one word is a stem in another (as in over and overdone). Many spelling checkers use affix normalization to reduce the size of the dictionary because, in theory, only word stems need to be stored. However, the differences between affixes and stems are blurred in many cases, and the rules for putting stem together with its affix have many exceptions. This technique introduces a source of possible error in the detection of misspelled words. This is because some misspelled words can be accepted as correct after they have been pared down to their so-called stems. Take the case of the word 'tailed', misspelled as 'talled'. Since 'ed' is an affix, it is removed before the word search. The resulting word stem 'tall' is found by the spelling checker to be right, even though the word 'talled' was wrong.

No affix normalization per se is used in this program. Rather, all word forms are included in the dictionary. As shown, affix stripping can lead to a decrease in the ability to correctly identify misspelled words. The dictionary has been compressed in such a way that similar patterns in words are not repeated needlessly, so the retention of most affixes will not greatly affect its size. The only affix which is stripped from all scanned words is " 's " which signifies possession. There would be little advantage in including in the dictionary all words which can have this affix since this group consists of all nouns and is much too large.

It is important to note the case (upper or lower) of the letters which make up a word before the dictionary check. 'Peter' and 'peter' could be different words and 'canada' is technically a misspelling. Each of 'the', 'The', and 'THE' is an acceptable word. But should all three be included in the dictionary? The solution consists of manipulating case so that one word is stored but all proper case forms of a word are accepted as correct words, and tagged as incorrect if the case is wrong. This problem of case is similar to the one of affixes, but is more general. There are really no rules which govern case other than those which address proper nouns and words which begin a sentence. The set of different case forms for all dictionary words is too large to include within the dictionary itself so case must be handled in another way.

Most commercial spelling programs simply ignore case differences, mapping all letters to the same case. With these programs, misspellings which result from the use of the wrong case of a letter are not detected. This is not a serious flaw because such misspellings make up a low percentage of the total set of misspelled words in a document. Moreover, they do not render a document unreadable. However, it would be preferable if some case checking were performed on the scanned words before dictionary checking.

Case analysis may be a little more important in a spelling program for scanned documents. A few pairs of upper and lower case letters are subject to being mistakenly exchanged by the scanner. 'P' and 'p' and 'O' and 'o' are examples of similarly shaped letter pairs which could be exchanged by the scanner.

The dictionary for this spelling program retains some case distinctions for proper nouns. For example, the name 'Betty' is stored in the dictionary with a capital 'B'. The name 'Bill' does not, on the other hand, appear, because the lower case 'bill', referring to an invoice, is present. In this instance, no mapping of letters occurs. To continue, every scanned word is given a class according to whether it is all lower case, leading upper case, or all upper case. The dictionary search for the word is then performed according to the class the word falls into.

After the final form of the word is determined by affix normalization and case analysis, the dictionary is searched for that form. The index is consulted first for the proper place to start the search. This is ~~going~~ to be the first byte of the section identified by the beginning letter of the word in question. The entire section is searched sequentially from this byte onward until either the word is matched, the end of the section is reached, or the word passes the place where it should have been alphabetically. If the word begins with an upper case letter, it may be found at either of two places in the dictionary. It could be found in the upper case portion if it is a proper name, or it could be found in the lower case portion if its leading letter has been capitalized for some other reason (e.g. it begins a sentence). In a situation like this, when the leading letter is upper case, if the word fails to be found in its existing form, it may still be in the dictionary, but in the lower case portion. Thus, the search is done again with the leading letter changed to its lower case equivalent (e.g. 'The' becomes 'the'). The search is performed until one of the end conditions is reached again. This type of search takes care of sentence capitals and proper names. Words consisting of all upper case letters are automatically mapped to lower case before the dictionary search. Words consisting of a mixture of upper and lower case letters must be matched exactly in the dictionary. Obviously, this is only a partial case analysis and will alleviate only some of the problems.

A word is only searched for in the dictionary if it has a chance of being there. Words of length of one are ignored by the spelling program because it is difficult to separate legal uses of one letter words from illegal uses. Although words such as 'a' and 'I' are correctly spelled words, they also can occur in a text produced from an optical scanner as the result of improperly recognized numbers, multiple errors, or noise (dirt on the paper). Other single letters can also occur in the text for the same reasons. Since it is difficult to determine whether or not they are correctly used in the text, they are simply ignored. This has the same effect as assuming that all one letter words are correct, but there is no dictionary search involved. Words of length longer than the longest dictionary word and words with illegal punctuation in them are immediately rejected as spelling errors without a dictionary search. The only punctuation character occurring in words in the dictionary is close quote/apostrophe (').

The spelling checker part of the program produces one file for the user called WRONG.WRD. This file contains a list of possible misspellings in the user's document. The list contains line numbers to help the user to find the misspelled words in context within their document files. The original input document file is unchanged. After checking the spelling, if the user specified both the checking and correcting parts of the spelling program, the program continues to the next step.

6. CORRECTION

The correction algorithm receives as input all words which the spelling checker cannot find in the dictionary. These words are assumed to be possible misspellings and to have at most one spelling error in them. This assumption is made by most commercial spelling programs. The correction process involves finding words which exist in the dictionary and are likely corrections of the misspelled word. The decision of the likelihood of being correct for each candidate word is based on how similarly spelled both words are to each other. Since an assumption of only one error is made and no transposition errors can occur, candidate words will differ from misspelled words in only one letter position.

After the list of candidate corrections is made, it is shown to the user along with the misspelled word. The user is given a choice of accepting the misspelled word as is, choosing the correct spelling from the candidate list, or correcting the word himself.

A popular algorithm for finding candidate corrections involves creating the candidate words from specific letter changes in the word. These candidate words can be found by systematically altering, one at a time, the letter positions of the misspelled word, trying to mimic the error process in reverse. Substitution errors are investigated by substituting for each letter in the misspelled word, all other letters in the alphabet, forming twenty-five new words for each letter position.

Deletion errors are investigated by deleting each letter in turn, to form one new word. For insertion errors, extra letters are added before, after, and between all existing letters of the misspelled word, to form a set of twenty-six new words for every new position. Each new word formed by one of these methods is checked against the dictionary and added to a list of candidate corrections if they are found there. This is the most general way in which most commercial spelling programs find new candidate words. This method has been altered and made more efficient in the spelling program implemented for this report.

It is clear from the previous discussion that the methods described could be time consuming. The time required can be reduced, however, by noting that some letter combinations will never occur. These combinations need not be checked for, thus saving time. Expanding on this idea, some letter combinations which occur with very low probability can also be cut out for further, larger time savings. The question becomes one of balancing between accuracy of the spelling correction and time consumption.

In the spelling program which I implemented, for all three types of errors, heuristic or statistical methods are used to determine probable corrections. These are based on the data analysis of a set of scanned test documents. This means that a set of scanned texts is used as a training set to determine which letter combinations have a low probability of occurrence in order to eliminate them from the correction processes. This is possible because the errors an optical scanner makes follow certain patterns in most cases. The choice of letter a scanner makes is based on an examination of the characteristics of each letter. If this choice is wrong, it is probably because the recognition algorithm of the optical scanner had difficulty discriminating between the characteristics of the true letter and the characteristics of the bad letter. This failure to distinguish letters is not random but consistent, and so a pattern forms in the mistakes made. For instance, the number '1' can be consistently picked instead of the true letter 'el' (or 'l'). This is a reliable pattern so that the most likely correction for a misspelling with a one '1' is found by substituting an el 'l' for it. The package takes advantage of the existence of such patterns and discards those transformations which are highly unlikely to occur and are very costly in time. Inevitably, sometimes such decisions will be wrong, and incorrect patterns will be mishandled. But the aim is for this to happen with very low probability. These few mistakes are worth making if program running time can be greatly reduced.

When dealing with substitution errors, each font is associated with a table called a confusion matrix. An example of a confusion matrix, for scanned letters 'a' through 'm', is shown in Table 1 on the following page. The confusion matrix indicates which letters in the text may be in error (vertical) as well as what the true letters for these errors may be (horizontal). The table is a 26 by 26 matrix where rows and columns are labelled by the letters of the alphabet. It is formed by noting which

Table 1. Example of a confusion matrix.

	TRUE LETTERS												
LETTER REPORTED BY SCANNER	a	b	c	d	e	f	g	h	i	j	k	l	m
a	811							1					2
b		158		1				16					1
c			255	2									
d				371			1						
e			1		1286		1						
f						190							
g							241						
h								537					1
i									521				
j										21			
k											156		
l								5				266	
m	1				7								232

letters in a test data set have been incorrectly scanned and by indicating in the matrix what letter was guessed instead and how many times this erroneous guess occurred.

In the example in Table 1, when one reads across the row labelled 'a', the numbers indicate that whenever the letter 'a' appeared in the scanned data, the correct letter was, in fact, an 'a' in 811 out of 814 times, an 'h' in 1 out of 814 times, and an 'm' in 2 out of 814 times. This table indicates to the correction program that the letter 'a' will probably be correct, for the test data, but that it could be an m or an h so it must check these letter substitutions but no others. Instead of twenty-five new possible words being formed, now only two are formed. The idea is to extrapolate from this test data analysis to all other texts which may be formed by scanning from this particular optical scanner. This spelling program assumes that for any document given to it from this scanner, the statistics in the confusion matrices are appropriate.

There are some important points to make about the correctness of this assumption:

(1) the test data set should be sufficiently large so that statistical variability is low and inferences about the larger world can be drawn from it,

(2) the text data should be representative of the material to be scanned, and

(3) these statistics necessarily pertain only to the optical scanner software for which they were compiled. Other scanner programs may produce different statistics.

In 'ordinary' correctors, each letter in a misspelled word is considered a possible mistake. So, each letter is replaced with each other letter in the alphabet and the dictionary is checked for the resulting word. The misspelling 'onf' could yield 'off' and 'one' from such a corrector. When confusion matrices are used, not all letters in a misspelled word are considered possible mistakes, only those which were incorrectly scanned in the training documents. In Table 1, the letter 'f' (of letters read by scanner) was actually an 'f' in all of the 190 times it appeared in the training documents. In this case, the letter 'f' is never considered as a possible mistake when it occurs in a seemingly misspelled word. Likewise, not all letters in the alphabet are considered as possible corrections. In other words, only certain letters are considered possible mistakes and they are corrected only by the letters paired with them in the table. Resulting words are then checked against the dictionary. If a word appears in the dictionary, it is added to the possible corrections list. This list is output from the correction program when it has finished correcting all words which were tagged as possible misspellings by the program's spelling checker.

As already stated, the use of a confusion matrix as an aid to spelling correction is a method which improves the correction algorithm over the 'brute force' method of trying all letters in the alphabet for all letters in the misspelled word. It represents a large savings in time, especially for large words. In the case of data produced from an optical scanner, it is the only way to deal with punctuation characters which have been mistaken for letters. Each punctuation character which occurs in the analysis of the test data has a row in the confusion matrix.

For deletion and insertion, the test data set was analysed to form a group of deletion letters and a group of insertion letters. In the case of deletion, the program simply inserts possible deletion letters into all possible positions in a misspelled word. In the case of insertion, the misspelled word is checked for each insertion letter and if one is found, it is deleted. The resulting words found from each process are checked against the dictionary and are added to the list of possible corrections if they are found. As an example, if it is found from an analysis of the test data that 'l' and 'w' make up a particular deletion group, but that 'r' does not, then the misspelled word 'ise' would add the words 'isle' and 'wise' from deletion errors to the correction list (if the words were in the dictionary). The word 'rise' would not be added because 'r' is not in the deletion group. If ''' were in the insertion group and the word 'ha'nd' was the misspelled word being corrected, the word 'hand' would be added to the correction list. As in the case for substitution errors, the statistics found from the test data set are expected to pertain to other documents which may be produced from the proper scanner program. This analysis should be good for most deletion/insertion patterns existing in documents but, of course, it will not be 100 per cent accurate. This is because all decisions are based on an analysis of test data. Although this data should be a good representation of the entire population, it is probably not an exact copy of the entire population and so cannot be 100 per cent accurate about all of the characteristics of such a population. For example, if the misspelled word 'ise' results from a deletion error of the letter 'r' in the general population but only rarely, this fact may not be discovered from the test data analysis. In this event, the word 'ise' can not be properly corrected. The statistical methods do not guarantee 100 per cent accuracy. However, processing is much faster when such methods are used and accuracy is still high.

There is another important assumption made when discussing the correction of possibly misspelled words. Misspelled words are assumed to contain only one error. This is an assumption made in most commercial spelling checkers and has been made for this spelling program as well. For texts typed by humans, about 80 per cent of misspelled words contain only one error [ref. 4]. Thus, the percentage of words which contain two or more errors is considered to be small enough to ignore when given the choice between the enormous number of possible corrections and the time consumption and accuracy of the spelling program. In the case of this spelling program, an analysis of the scanned test documents of approximately 2886 words reveals that the per cent of misspelled words which contain only one error is about 77 (see Tables 5a, 5b, 5c, 5d, and 5e). It appears that this is still a good assumption to make.

There are two major problems with checking for multiple errors which make it inefficient. The time it takes to check a word for multiple errors is more than triple that of just checking for single errors, based on an experiment of the test documents. This is because all words found from single transformations are themselves transformed again (and for as many errors as needed to check for) leading to a huge increase in the number of words checked against the dictionary. Since most of these words will be wrong anyway, it really is a waste of time to check for the few that would be correct. In some cases, it would take less time to go through the document manually! Also, although most of the words checked against the dictionary are incorrect, there is still an increase in the number of words added to the corrections list. Since the point of such a list is to indicate probable corrections to the user, the inclusion of too many words renders the list less useful because too many words can be confusing.

The correction program does more than just output a list of candidate corrections for the user. A version of the corrected text is automatically generated and output as well. As far as can be determined, this is not done by commercial programs. Commercial programs are interactive, giving the user an opportunity to correct misspelled words as the program finds them. In this way, corrected versions of the text are produced but they are the result of direct user intervention. If the user has no inclination to correct the words, then they are not corrected by the program.

This spelling program automatically produces a corrected text for the user. The choice of correct form for the misspelled word is chosen from its list of candidate corrections. If a misspelled word has no candidate corrections, there is no final correction and the word appears in the corrected text exactly as it appeared in the original text.

For the implemented spelling program, when a candidate word is added to the correction list for a misspelled word, a ranking is computed as well. The candidate word with the highest ranking is chosen as the final correction for the misspelled word in question.

The ranking of a word is formed from the data analysis of the test data set. For substitution errors, the most common substitution error for each misspelled letter was noted, then the next-most common and so on.

When a new candidate word is formed, it receives the ranking of the letter which was used to produce it. All words formed from reversing substitution errors have higher ranks than words formed from reversing deletion errors which, in turn, have a higher ranking than words formed from reversing insertion errors. The rationale behind this is that substitution errors by far make up the bulk of the errors which cause scanner read misspelled words. For the test document set, at least 57 per cent (see Tables 5a - 5e) of the misspelled words resulted from substitution errors. Deletion errors come next, for at most 4 per cent, and insertion errors accounted for at most 1 per cent of the errors. So, a rough ranking would divide the words in the candidate list into three ranked parts as (1) substitution, (2) deletion, and (3) insertion.

At the next level, words within each group are given a ranking, based on the occurrence value of the different scanned letters within the word. Although this ranking ultimately depends on the confusion matrices found by the data analysis, it also takes into account other factors. Since misspelled words with punctuation marks within them are always considered misspelled, all these words have a higher ranking than words without punctuation in them. Letters (and thus words) which are lower case letters have a higher ranking than letters which are upper case letters. This is because lower case letters occur more often and are more likely to be correct. For instance, if the character '!' had both 'L' and 'l' as possible character corrections, both occurring as the true correction in 5 per cent of the occurrences, 'l' would still be given a higher ranking than 'L'. In this way, ties are broken by noting which character would make the most sense most of the time. When this kind of judgment cannot be made, ties are broken randomly. Admittedly, some of these ranking decisions are a bit arbitrary. Most ranking decisions are based on the probabilistic analysis of the test set but some of it is based on notions of characteristics of the English language. The purpose is to make sure a correct word will be chosen as often as possible. This is , in the end, an extremely difficult thing to do. Moving aside the more arbitrary decisions for a moment, it means that when a word is picked because it contains a character correction which occurs more often than any other character correction, it will be chosen 100 per cent of the time even though it only occurred say 40 per cent of the time. This means that this character pattern will only be corrected properly 40 per cent of the time. If all the words were correctable, and if they all had at least one correction as high as 40 per cent, then the entire corrected text would have only 40 per cent of its misspelled words corrected. But it has been found in this research project that only about 77 per cent of the misspelled words in a document are correctable. Also, many of the correctable words do not have letter corrections which are as strong as 40 per cent. The longer the candidate list, the weaker (the lower the percentage of any one word) each individual candidate correction is.

The calculation to determine which candidate correction to choose is tricky. Rankings can change slightly from one document to the next, even though the statistics do not change much. Where one word is chosen as a correction based on rankings from the test data, it will also be chosen for all other documents using the same spelling program, even though it may not be the correct choice in such cases. The statistical line between what is chosen and what is not when two words have close rankings is hard to draw. This problem can only be corrected by monitoring the system continuously to see what kind of adjustments should be made, if any can be made. The automatically corrected text may never enjoy a high percentage of corrected misspelled words simply because it is just too hard to rank candidate corrections. There are too many factors involved.

7. DATA COLLECTION

The data used to develop this program's heuristics was compiled from various sources. A list of these are as follows:

(1) the top 499 words taken from a list of most common words in the English language [ref. 15],

(2) a group of words called Spondee words, normally used in the analysis of ** [ref. 17],

(3) a group of phonetically balanced words [ref. 17],

(4) six sets of ten sentences each from a group of phonetically balanced sentences [ref. 16],

(5)ten sets of each alphabetic character, and

(6)six pages from passages taken from various school texts [ref. 17].

These texts together supplied 2886 words and 10,576 letters to the test texts and represent a good sample from the virtually infinite population of everyday English. The exact distributions of letters and words is shown in Tables 2 and 3.

This spelling checker/corrector will work on any text file. However, it has been designed for the textual output produced by an optical scanner. One variable the scanner needs to know about the textual image is the font it was printed in. The font type governs the printed shape and size of the character set. Some fonts include OCR A (Optical Character Recognition A), OCR B,

Table 2. The number of words in each document used for pre-tests.

document contents	document id	number of words
the ranked list	rnklst	499
the spondee words	spnd	74
the phonetically balanced sentences	phnsen	478
the phonetically balanced words	phnwrd	208
the character sets	stat	521
the school texts	schtxt	1106
total number of words		2886

2.

Table 3. Letter frequency in each document used for pre-tests.

	a	b	c	d	e	f	g	h	i	j	k	l	m
rnklst	180	39	74	84	322	48	59	109	138	4	21	126	85
spnd	56	18	14	36	44	2	14	38	22	0	16	26	14
phnsen	131	28	51	73	270	50	43	138	95	2	33	84	28
phnwrd	66	10	18	26	106	10	10	42	42	4	10	32	20
stat	12	10	11	10	11	10	10	11	10	10	10	10	10
schtxt	372	58	89	157	545	74	127	294	242	5	68	177	93
totals	817	163	257	386	1298	194	263	632	549	25	158	455	250

	n	o	p	q	r	s	t	u	v	w	x	y	z
rnklst	165	203	45	2	155	148	191	77	32	57	5	52	0
spnd	18	62	12	0	52	34	30	18	0	26	0	14	0
phnsen	100	140	36	1	108	149	159	48	18	39	2	25	5
phnwrd	44	52	8	0	36	46	64	18	12	36	0	16	0
stat	10	10	10	10	12	11	11	10	10	10	10	10	10
schtxt	282	306	58	5	258	262	386	84	25	114	15	72	0
totals	619	773	169	18	621	650	841	255	97	282	32	189	15

	A	B	C	D	E	F	G	H	I	J	K	L	M
rnklst	1	0	0	0	1	0	0	1	2	0	1	1	0
spnd	0	0	0	2	2	0	0	0	0	0	0	2	0
phnsen	7	2	0	0	0	1	1	5	2	0	1	8	3
phnwrd	3	1	2	2	2	0	0	1	2	0	0	8	0
stat	10	10	11	10	10	10	10	10	10	10	10	10	10
schtxt	17	2	6	7	0	2	1	12	27	1	0	1	12
totals	38	15	19	21	15	13	12	29	43	11	12	30	25

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
rnklst	1	0	0	0	1	1	2	0	0	0	0	0	0
spnd	1	2	1	0	1	2	0	0	0	1	0	0	0
phnsen	1	0	3	0	3	4	30	0	0	2	0	0	0
phnwrd	2	2	1	0	1	2	2	0	0	1	0	1	0
stat	10	10	10	10	10	10	10	10	10	10	10	10	10
schtxt	4	7	9	4	0	13	23	0	0	14	0	3	0
totals	19	21	24	14	16	32	67	10	10	28	10	14	10

ELITE, COURIER, LETTER GOTHIC, and PICA. The scanner used has software which distinguishes between fonts. For different fonts, the scanner uses different recognition algorithms. This is because the same letter in two different fonts may look different to the scanner. If a text file printed in COURIER is scanned by the recognition algorithm for ELITE, the rate of recognition will be low. The optical scanner software used produces output from text printed in five different types of fonts. These are OCR A, OCR B, COURIER, ELITE, and GOTHIC. The character set printed in these five different fonts is shown in Table 4.

The difference in font leads to different characteristics in the misspelled word as well. For example, texts printed in OCR A usually have fewer mistakes in the scanned output than those printed in any of the other fonts. This is because the OCR A font was developed for use with optical scanners and all the characters have been designed with distinguishing characteristics. Each font has a different set of mistaken characters. A spelling program must take this into account and have parallel algorithms to deal with each font.

The original set of documents was printed in the five different fonts of OCR A, OCR B, COURIER, ELITE, and GOTHIC. The data for the fonts COURIER, ELITE, and GOTHIC were printed from a laser printer, the H/P LaserWriter. The data for the fonts OCR A and OCR B were printed from a daisy wheel printer.

The five sets of documents were scanned by an optical scanner using software developed at Concordia University. Five

Table 4. Examples of font character sets.

OCR A:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

OCR B:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

COURIER:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

ELITE:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

GOTHIC

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z sets of scanned data output were produced, one scanned set for each font.) These data are used to develop and test the checking and correcting programs.

The documents used to form the training set seem to be relatively representative of common English texts. In order to have confidence in the performance of the confusion matrices on other texts produced from this scanner software, the training documents must be sufficiently large to enable precise estimates of the confusion matrices. The binomial probability distribution is used to estimate the precision of the proportions in the confusion matrices. To determine the (1 - α) confidence interval around the true proportion, the following formula is used:

$$\{P' - Z_{\alpha/2}[P(1-P)/N]^{.5},\ P' + Z_{\alpha/2}[P(1-P)/N]^{.5}\}$$

where N is the number of occurrences,

P' is the observed proportion,

P is the true proportion, and

$Z_{\alpha/2}$ is the point on the standard curve with tail probability of α/2.

A confidence interval of X per cent around the true proportion means that the true proportion will be within the interval X per cent of the time. A 95 per cent level of confidence is the standard. When N is taken to be 96 and P to be .5, the 95 per cent confidence interval has a half-length of 0.1. For P much higher or lower than .5, the interval is narrower. Also, if N, the sample size, is larger, a narrower confidence interval is obtained. For an interval half-length of .05, almost

400 occurrences of each letter must be sampled. For the more common letters (a,e,h,i,l,n,o,r,s,t), this is already the case. However, to get so many occurrences for all letters, many more documents need to be analysed.

Thus, the texts provide 95 per cent confidence that the entries for all but the most uncommon letters have a narrow range of statistical variation (less than 10 per cent). This means that at least 96 occurrences of each letter must be sampled (i.e. must occur in the training documents) for most of the estimates in the confusion matrices to be correct within a 10 per cent range. In Table 3, it can be seen that all the lower case letters except 'j', 'q', 'x', 'z' meet these criteria. No upper case letters do. Since these letters are uncommon, errors occurring among them are rare.

It is of interest to note for each set of documents how many substitution, insertion, and deletion errors were made. This information, along with the confusion matrices, was used to produce the ranks of the candidate corrections for each misspelled word, as described in a previous section. Another variable of interest is the split word calculation. In the process of scanning these documents, the scanner produces a difficult kind of error to deal with which can be described as an insertion error where the character being inserted is a space. This results in split words, most of which are found to be misspelled words by the spelling checker. These split words are very hard to correct because each part of the word is taken to be one complete word by the program and thus will usualy have more than one spelling error. These spelling errors can be seen to be deletion errors and the number of deletion errors will equal the number of letters of the original word which have been split off from the part being examined. Tables 5a - 5e give this information for each font along with the number of actual mistaken words appearing in each set of documents and words having more than one error in them. This last count does not include any split words unless they also have other errors occurring in them. Most calculations are done with the total number of words in a document which have a length greater than one character. This total is given in the tables along with the total number of words in a document (regardless of length) for comparison. Also in these tables, incorrect words are broken into five distinct categories, substitution, deletion, insertion, split words, and words which contain greater than one error.

This set of tables gives an overall evaluation of the optical scanner software and its performance on each font. Misspellings occur at the rates of 11% for OCR A, 31% for OCR B, 35% for COURIER, 27% for ELITE, and 10% for GOTHIC. The fonts with the best performance seem to be OCR A and Gothic. It was expected that the OCR A font would do well because it was developed to be read by optical scanners. However, the other OCR font, OCR B, does not do as well. COURIER has a slightly worse performance than the other three fonts of COURIER, OCR B, and ELITE.

Table 5a. Analysis of scanned document sets for OCR A.

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
rnklst	499	497	424	73
spnd	74	74	64	10
phnsen	478	465	384	81
phnwrd	208	208	165	43
stat	521	1	1	0
schtxt	1106	1074	1021	53
totals	2886	2319	2059	260

document id	incorrect					
	total	sub	ins	del	split	>1 err
1 rnklst	73	60	0	0	0	13
2 spnd	10	6	0	0	0	4
3 phnsen	81	32	0	0	41	8
4 phnwrd	43	21	0	0	13	9
5 stat	0	0	0	0	0	0
6 schtxt	53	34	0	0	0	19
totals	260	153	0	0	54	53

Table legend:

sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

Table 5b. Analysis of scanned document sets for OCR B.

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
rnklst	499	497	332	165
spnd	74	74	45	29
phnsen	478	465	261	204
phnwrd	208	208	109	99
stat	521	1	0	1
schtxt	1106	1074	844	230
totals	2886	2319	1591	728

document id	incorrect					
	total	sub	ins	del	split	>1 err
rnklst	165	121	0	0	5	39
spnd	29	20	0	0	0	9
phnsen	204	70	0	0	80	54
phnwrd	99	35	0	0	27	37
stat	1	1	0	0	0	0
schtxt	230	193	0	0	14	23
totals	728	440	0	0	126	162

Table legend:

sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

Table 5c. Analysis of scanned document sets for COURIER.

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
rnklst	499	497	324	173
spnd	74	74	37	37
phnsen	478	465	320	145
phnwrd	208	208	144	64
stat	521	1	1	0
schtxt	1106	1074	682	392
totals	2886	2319	1508	811

document id	incorrect					
	total	sub	ins	del	split	>1 err
rnklst	173	136	0	0	0	37
spnd	37	17	0	0	0	20
phnsen	145	127	0	0	0	18
phnwrd	64	49	0	0	0	15
stat	0	0	0	0	0	0
schtxt	392	290	0	0	3	99
totals	811	619	0	0	3	189

Table legend:
sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

Table 5d. Analysis of scanned document sets for ELITE.

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
rnklst	499	497	312	185
spnd	74	74	37	37
phnsen	478	465	371	94
phnwrd	208	208	155	53
stat	521	1	1	0
schtxt	1106	1074	818	256
totals	2886	2319	1694	625

document id	incorrect					
	total	sub	ins	del	split	>1 err
rnklst	185	101	1	17	0	66
spnd	37	18	0	0	0	19
phnsen	94	61	1	2	0	30
phnwrd	53	38	0	0	0	15
stat	0	0	0	0	0	0
schtxt	256	142	5	2	1	106
totals	625	360	7	21	1	236

Table legend:

sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

Table 5e. Analysis of scanned document sets for GOTHIC

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
rnklst	499	497	429	68
spnd	74	74	63	11
phnsen	478	465	432	33
phnwrd	208	208	188	20
stat	521	1	1	0
schtxt	1106	1074	981	93
totals	2886	2319	2094	225

document id	incorrect					
	total	sub	ins	del	split	>1 err
rnklst	68	53	0	0	1	14
spnd	11	7	0	0	0	4
phnsen	33	28	0	0	2	3
phnwrd	20	20	0	0	0	0
stat	0	0	0	0	0	0
schtxt	93	84	0	0	4	5
totals	225	192	0	0	7	26

Table legend:

sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

For all five fonts, substitution errors were the most common, followed by those errors which occur more than once in a word. Insertion and deletion errors occur very rarely. This would suggest that spelling programs which adopted a strategy in which substitution errors were corrected, insertion and deletion errors were ignored, and some attempt were made to correct multiple errors, could have efficient correction rates. Finally, words which are split by the scanner software are a problem for some of the fonts. Some correction of these might also be desirable.

8. PRE-TEST ANALYSIS

The texts used to construct the heuristics and the confusion matrices for the spelling program can be called the pre-test or training set. These pre-test texts were analysed by hand for incorrectly recognized letters. The use of heuristics has already been explained. The confusion matrices used by the program for the five fonts can be found in appendix A.

According to the pre-test analysis, no font has a confusion matrix with more than 43 rows (the characters identified wrongly by the scanner). The largest set of possible correct letters for a single row is 9. ELITE has the largest matrix and OCR A has the smallest. This indicates that ELITE makes the largest number of unique errors in scanning and OCR A makes the least.

The deletion algorithm is run on texts from OCR B data. The insertion algorithm is run on texts from OCR A, COURIER, and ELITE data. Of course, each font has its own table of probable insertion and deletion errors. Even though some fonts do not appear to have insertion or deletion errors, sometimes a split word or a multiple error will mimic a simpler error. If these things were noticed as some kind of pattern in the preliminary analysis of the documents, then insertion and deletion groups could be formed. GOTHIC was not found to have any significant deletion or insertion errors in the pre-test analysis.

It is useful to note how the analysis of the pre-test texts affected the spelling check and correction of the same pre-test texts. For this purpose, a number of variables were measured both after checking the documents for incorrectly spelled words and after the correction of these words.

The analysis done on the data after the spelling check is summarized in Tables 6a - 6e and Tables 7a - 7e. These tables show how well the spelling checker performed on the 2886 words for each font. To know how well the checker performed, it is not enough to count just the number of mistakes the checker found. The checker could have made two serious errors in the process of listing incorrect words:

(I) It could fail to identify some incorrect words appearing in the document as incorrect (Tables 7a-7e). This can happen in four cases:

(1) When a word is misspelled as another word which happens to be in the dictionary. For example, ~~the word~~ 'from' could be misspelled as 'form'.

(2) When the first letter in a word is misrepresented as its upper case equivalent. For example, 'perhaps' is misspelled as 'Perhaps'.

(3) When a misspelled word is only one letter long.

(4) When a word is split into two or more parts each having a length of one letter.

(II) The spelling checker could also identify as incorrect some words which are actually correct (Tables 6a-6e). This happens in the case where a word does not exist in the dictionary but it is still a correct word. This happens most often with proper nouns (such as Kimba) and rarely used words (such as some chemical compounds).

Tables 6a - 6e and 7a - 7e give the counts for both kinds of mistakes as well as for the words which were identified correctly as misspelled and spelled right. Tables 6a - 6e give the statistics showing which correct words were correctly identified as such. Tagged words are assumed by the spelling program to be incorrect, because these words were not found in its dictionary. Examples of tagged words are '1n' (numeral one instead of letter 'i'), 'Thc' (the letter 'c' instead of the letter 'e'), and 'pancaKe' (upper case 'K' instead of lower case 'k'). Thus, untagged words are the words which were properly identified. Tables 7a - 7e give the breakdown of spelling errors which occurred in the training or pre-test documents. They give this breakdown both for errors which were detected (t) and for errors which were not detected (n).

When the document is checked for spelling errors, the difference in font is immaterial. Correct words always have the same definition, namely that they are two or more letters long and contain only alphabetic characters or the close quote/apostrophe. Likewise, the overall design of the correction package remains the same for all fonts. However, the individual particulars for their correction algorithms will differ between the fonts.

Table 6a. OCR A data analysis for correct words of pre-test group after spelling check.

document id	total	correct tagged	correct not tagged
rnklst	424	0	424
spnd	64	6	58
phnsen	384	0	384
phnwrd	165	0	165
stat	1	0	1
schtxt	1021	44	977
totals	2059	50	2009

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 6b. OCR B data analysis for correct words of pre-test group after spelling check.

document id	total	correct tagged	correct not tagged
rnklst	332	0	332
spnd	45	4	41
phnsen	261	0	261
phnwrd	109	0	109
stat	0	0	0
schtxt	844	1	843
totals	1591	5	1586

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 6c. COURIER data analysis for correct words of pre-test group after spelling check.

document id	total	correct tagged	correct not tagged
rnklst	324	0	324
spnd	37	3	34
phnsen	320	0	320
phnwrd	144	0	144
stat	1	0	1
schtxt	682	26	656
totals	1508	29	1479

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 6d. ELITE data analysis for correct words of pre-test group after spelling check.

document id	total	correct tagged	not tagged
rnklst	312	0	312
spnd	37	2	35
phnsen	371	3	368
phnwrd	155	0	155
stat	1	0	1
schtxt	818	41	777
totals	1694	46	1648

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 6e. GOTHIC data analysis for correct words of pre-test group after spelling check.

document id	total	correct tagged	correct not tagged
rnklst	429	0	429
spnd	63	4	59
phnsen	432	3	429
phnwrd	188	0	188
stat	1	0	1
schtxt	981	33	948
totals	2094	40	2054

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 7a. OCR A data analysis for incorrect words in pre-test group after spelling check.

document	sub			ins			del		
id	tot	yes	no	tot	yes	no	tot	yes	no
rnklst	60	57	3	0	0	0	0	0	0
spnd	6	6	0	0	0	0	0	0	0
phnsen	32	31	1	0	0	0	0	0	0
phnwrd	21	18	3	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0
schtxt	34	34	0	0	0	0	0	0	0
totals	153	146	7	0	0	0	0	0	0

document	split			>1 err		
id	tot	yes	no	tot	yes	no
rnklst	0	0	0	13	11	2
spnd	0	0	0	4	4	0
phnsen	41	34	7	8	8	0
phnwrd	13	12	1	9	8	1
stat	0	0	0	0	0	0
schtxt	0	0	0	19	18	1
totals	54	46	8	53	49	4

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

Table 7b. OCR B data analysis for incorrect words in pre-test group after spelling check.

document	sub			ins			del		
id	tot	yes	no	tot	yes	no	tot	yes	no
rnklst	121	118	3	0	0	0	0	0	0
spnd	20	20	0	0	0	0	0	0	0
phnsen	70	68	2	0	0	0	0	0	0
phnwrd	35	32	3	0	0	0	0	0	0
stat	1	1	0	0	0	0	0	0	0
schtxt	193	171	22	0	0	0	0	0	0
totals	440	410	30	0	0	0	0	0	0

document	split			>1 err		
id	tot	yes	no	tot	yes	no
rnklst	5	5	0	39	38	1
spnd	0	0	0	9	9	0
phnsen	80	65	15	54	48	6
phnwrd	27	23	4	37	36	1
stat	0	0	0	0	0	0
schtxt	14	14	0	23	23	0
totals	126	107	19	162	154	8

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

Table 7c. COURIER analysis for incorrect words in pre-test group after spelling check.

document id	sub tot	sub yes	sub no	ins tot	ins yes	ins no	del tot	del yes	del no
rnklst	136	132	4	0	0	0	0	0	0
spnd	17	16	1	0	0	0	0	0	0
phnsen	127	123	4	0	0	0	0	0	0
phnwrd	49	45	4	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0
schtxt	290	278	12	0	0	0	0	0	0
totals	619	594	25	0	0	0	0	0	0

document id	split tot	split yes	split no	>1 err tot	>1 err yes	>1 err no
rnklst	0	0	0	37	37	0
spnd	0	0	0	20	20	0
phnsen	0	0	0	18	18	0
phnwrd	0	0	0	15	15	0
stat	0	0	0	0	0	0
schtxt	3	2	1	99	98	1
totals	3	2	1	189	188	1

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

Table 7d. ELITE data analysis for incorrect words in pre-test group after spelling check.

document	sub			ins			del		
id	tot	yes	no	tot	yes	no	tot	yes	no
rnklst	101	94	7	1	1	0	17	9	8
spnd	18	17	1	0	0	0	0	0	0
phnsen	61	60	1	1	1	0	2	2	0
phnwrd	38	34	4	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0
schtxt	142	136	6	5	5	0	2	1	1
totals	360	341	19	7	7	0	21	12	9

document	split			>1 err		
id	tot	yes	no	tot	yes	no
rnklst	0	0	0	66	60	6
spnd	0	0	0	19	19	0
phnsen	0	0	0	30	30	0
phnwrd	0	0	0	15	15	0
stat	0	0	0	0	0	0
schtxt	1	1	0	106	104	2
totals	1	1	0	236	228	8

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.

no = misspelled word is not flagged.

Table 7e. GOTHIC data analysis for incorrect words in pre-test group after spelling check.

document	sub			ins			del		
id	tot	yes	no	tot	yes	no	tot	yes	no
rnklst	53	44	9	0	0	0	0	0	0
spnd	7	7	0	0	0	0	0	0	0
phnsen	28	24	4	0	0	0	0	0	0
phnwrd	20	19	1	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0
schtxt	84	70	14	0	0	0	0	0	0
totals	192	164	28	0	0	0	0	0	0

document	split			>1 err		
id	tot	yes	no	tot	yes	no
rnklst	1	1	0	14	13	1
spnd	0	0	0	4	4	0
phnsen	2	1	1	3	2	1
phnwrd	0	0	0	0	0	0
stat	0	0	0	0	0	0
schtxt	4	4	0	5	5	0
totals	7	6	1	26	24	2

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

The correction part of the spelling package consists of five groups of algorithms, one for each font. The flow of these algorithms is basically the same. Spelling errors produced by optical scanners break down into 3 types (as opposed to the 4 normal ones): substitution, insertion, and deletion. Substitution errors are by far the most prevalent. All five fonts are checked for substitution errors. Only OCR A, COURIER, and ELITE are checked for insertion errors. Only OCR B is checked for deletion errors. The misspelled word is first checked for substitution errors, then for insertion and deletion errors if indicated for the font. All the other combinations of font and error were not found to occur significantly in the pre-test analysis.

Tables 8a - 8e show how well these heuristics performed for the correction part of the spelling program. These tables indicate how many words were properly corrected in the corrected version of the scanned documents. They also show how many words had the true word occurring in their candidate correction list (add the number of properly corrected words to the number of words not corrected but with the right word in its candidate list). These tables show how well the correction algorithms did for the pre-test documents.

Substitutions were generally corrected. Insertion and deletion (almost never occurred), split words, and multiple errors (>1 err) were generally not corrected. Since correction algorithms for split words and multiple errors were not provided,

Table 8a. OCR A data analysis for incorrect words in pre-test group after spelling correction.

document id	sub				ins				del			
	tot	cor	not		tot	cor	not		tot	cor	not	
			c	n			c	n			c	n
rnklst	57	53	2	2	0	0	0	0	0	0	0	0
spnd	6	5	0	1	0	0	0	0	0	0	0	0
phnsen	31	28	3	0	0	0	0	0	0	0	0	0
phnwrd	18	14	3	1	0	0	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0	0	0	0
schtxt	34	30	2	2	0	0	0	0	0	0	0	0
totals	146	130	10	6	0	0	0	0	0	0	0	0

document id	split				>1 err			
	tot	cor	not		tot	cor	not	
			c	n			c	n
rnklst	0	0	0	0	11	0	0	11
spnd	0	0	0	0	4	0	0	4
phnsen	34	3	0	31	8	0	0	8
phnwrd	12	1	0	11	8	0	0	8
stat	0	0	0	0	0	0	0	0
schtxt	0	0	0	0	18	0	0	18
totals	46	4	0	42	49	0	0	49

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words were corrected properly in the 'corrected' text.

not = misspelled words were not corrected properly but [c] were included in the list of candidates or [n] were not in the list.

Table 8b. OCR B data analysis for incorrect words in pre-test group after spelling correction.

document	sub				ins				del			
id	tot	cor	not		tot	cor	not		tot	cor	not	
			c	n			c	n			c	n
rnklst	118	111	3	4	0	0	0	0	0	0	0	0
spnd	20	18	0	2	0	0	0	0	0	0	0	0
phnsen	68	68	0	0	0	0	0	0	0	0	0	0
phnwrd	32	28	2	2	0	0	0	0	0	0	0	0
stat	1	1	0	0	0	0	0	0	0	0	0	0
schtxt	171	158	4	9	0	0	0	0	0	0	0	0
totals	410	384	9	17	0	0	0	0	0	0	0	0

document	split				>1 err			
id	tot	cor	not		tot	cor	not	
			c	n			c	n
rnklst	5	0	1	4	38	0	0	38
spnd	0	0	0	0	9	0	0	9
phnsen	65	0	2	63	48	0	0	48
phnwrd	23	3	3	17	36	0	0	36
stat	0	0	0	0	0	0	0	0
schtxt	14	1	2	11	23	0	0	23
totals	107	4	8	95	154	0	0	154

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words were corrected properly in the 'corrected' text.
not = misspelled words were not corrected properly but [c] were included in the list of candidates or [n] were not in the list.

Table 8c. COURIER data analysis for incorrect words in pre-test group after spelling correction.

document id	sub				ins				del			
	tot	cor	not c	not n	tot	cor	not c	not n	tot	cor	not c	not n
rnklst	132	125	5	2	0	0	0	0	0	0	0	0
spnd	16	16	0	0	0	0	0	0	0	0	0	0
phnsen	123	123	0	0	0	0	0	0	0	0	0	0
phnwrd	45	42	3	0	0	0	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0	0	0	0
schtxt	278	255	9	14	0	0	0	0	0	0	0	0
totals	594	561	17	16	0	0	0	0	0	0	0	0

document id	split				>1 err			
	tot	cor	not c	not n	tot	cor	not c	not n
rnklst	0	0	0	0	37	0	0	37
spnd	0	0	0	0	20	0	0	20
phnsen	0	0	0	0	18	0	0	18
phnwrd	0	0	0	0	15	0	0	15
stat	0	0	0	0	0	0	0	0
schtxt	2	0	0	2	98	0	0	98
totals	2	0	0	2	188	0	0	188

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words were corrected properly in the 'corrected' text.
not = misspelled words were not corrected properly but [c] were included in the list of candidates or [n] were not in the list.

Table 8d. ELITE data analysis for incorrect words in pre-test group after spelling correction.

document id	sub				ins				del			
	tot	cor	not		tot	cor	not		tot	cor	not	
			c	n			c	n			c	n
rnklst	94	89	1	4	1	0	0	1	9	0	0	9
spnd	17	17	0	0	0	0	0	0	0	0	0	0
phnsen	60	53	4	3	1	0	0	1	2	0	0	2
phnwrd	34	29	5	0	0	0	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0	0	0	0
schtxt	136	116	10	10	5	1	0	4	1	0	0	1
totals	341	304	20	17	7	1	0	6	12	0	0	12

document id	split				>1 err			
	tot	cor	not		tot	cor	not	
			c	n			c	n
rnklst	0	0	0	0	60	0	0	60
spnd	0	0	0	0	19	0	0	19
phnsen	0	0	0	0	30	0	0	30
phnwrd	0	0	0	0	15	0	0	15
stat	0	0	0	0	0	0	0	0
schtxt	1	0	0	1	104	2	0	102
totals	1	0	0	1	228	2	0	226

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words were corrected properly in the 'corrected' text.
not = misspelled words were not corrected properly but [c] were included in the list of candidates or [n] were not in the list.

Table 8e. GOTHIC analysis for incorrect words in pre-test group after spelling correction.

document	sub				ins				del			
id	tot	cor	not		tot	cor	not		tot	cor	not	
			c	n			c	n			c	n
rnklst	44	40	1	3	0	0	0	0	0	0	0	0
spnd	7	7	0	0	0	0	0	0	0	0	0	0
phnsen	24	24	0	0	0	0	0	0	0	0	0	0
phnwrd	19	17	1	1	0	0	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0	0	0	0	0
schtxt	70	58	2	10	0	0	0	0	0	0	0	0
totals	164	146	4	14	0	0	0	0	0	0	0	0

document	split				>1 err			
id	tot	cor	not		tot	cor	not	
			c	n			c	n
rnklst	1	0	0	1	13	0	0	13
spnd	0	0	0	0	4	0	0	4
phnsen	1	0	0	1	2	0	0	2
phnwrd	0	0	0	0	0	0	0	0
stat	0	0	0	0	0	0	0	0
schtxt	4	0	0	4	5	0	0	5
totals	6	0	0	6	24	0	0	24

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words were corrected properly in the 'corrected' text.
not = misspelled words were not corrected properly but [c] were included in the list of candidates or [n] were not in the list.

their poor correction performance is not surprising. When they are corrected, it is because their effect on the misspelled words behaved like one of substitution, insertion, or deletion.

In the end, what really matters is the overall performance of the spelling program system. Tables 9a - 9e measure this statistic as a system summary. The variable system efficiency is used in this summary, System efficiency measures how well the spelling program performed by calculating how much better (i.e., how much more correct) the corrected version of the scanned text is over the scanned text itself. It is calculated as a ratio. The numerator is the difference in the number of correct words in the corrected and scanned documents and the denominator is the number of errors present in the version of the scanned document. This ratio gives the percentage of improvement in the spelling, which is a measurement of how well the spelling program operates. If all incorrect words are corrected, efficiency is 100 per cent. If none are corrected, the efficiency is 0 per cent.

While the system efficiency percentages vary from font to font, they are all within 20 per cent of each other. There is no one font where efficiency is particularly low or particularly high. This shows that the implemented spelling program is fairly stable in its operations and performs with the same basic productivity for all five fonts. Even though OCR A and GOTHIC are fonts that scan well, in terms of total system efficiency, they are about the same as the others. However, if correctness

Table 9a. System summary for OCR A.

document id	total # words w/ >1 letters	scanned document # words correct	corrected document # words correct	system efficiency
rnklst	497	424	477	73%
spnd	74	64	69	50%
phnsen	465	384	415	38%
phnwrd	208	165	180	35%
stat	1	1	1	***
schtxt	1074	1021	1051	57%
totals	2319	2059	2193	52%

Table 9b. System summary for OCR B.

document id	total # words w/ >1 letters	scanned document # words correct	corrected document # words correct	system efficiency
rnklst	497	332	443	67%
spnd	74	45	63	62%
phnsen	465	261	329	33%
phnwrd	208	109	140	31%
stat	1	0	1	100%
schtxt	1074	844	1003	69%
totals	2319	1591	1979	53%

Table 9c. System summary for COURIER.

document id	total # words w/ >1 letters	scanned document ------ # words correct	corrected document -------- # words correct	system efficiency
rnklst	497	324	449	72%
spnd	74	37	53	43%
phnsen	465	320	443	85%
phnwrd	208	144	186	66%
stat	1	1	1	***
schtxt	1074	682	937	65%
totals	2319	1508	2069	69%

Table 9d. System summary for ELITE.

document id	total # words w/ >1 letters	scanned document ------- # words correct	corrected document --------- # words correct	system efficiency
rnklst	497	312	401	48%
spnd	74	37	54	46%
phnsen	465	371	424	56%
phnwrd	208	155	184	55%
stat	1	1	1	***
schtxt	1074	818	937	46%
--------				
totals	2319	1694	2001	49%

Table 9e. System summary for GOTHIC.

		scanned document	corrected document	
document id	total # words w/ >1 letters	# words correct	# words correct	system efficiency
rnklst	497	429	469	59%
spnd	74	63	70	64%
phnsen	465	432	456	73%
phnwrd	208	188	205	85%
stat	1	1	1	***
schtxt	1074	981	1039	62%
totals	2319	2094	2240	65%

in scanning and spelling are both examined, GOTHIC has a better performance than all the others so far.

Some care must be taken in evaluating a system on the same data from which its statistical analysis was performed. The analysis will always be most true for the data from which it was drawn, so the resulting spelling program probably performs very well on this data. The point of the statistical analysis is for the methods developed on the pre-test data to be transferable to other data as well, as long as this data is part of the same statistical environment. This can only be done if the pre-test sample was chosen and analysed properly.

External tests which compare the spelling program with other spelling programs are another way of evaluating statistical methods used to build the implemented spelling program. These tests will hopefully show how well the characteristics of other document groups parallel those of the sample group.

9. COMPARISON OF SPELLING PROGRAMS

The analysis of the spelling program developed for this project is not complete until it includes some performance statistics for the program compared with performance statistics for other commercial spelling programs. This program was compared to three commercial spelling programs, PAPERBACK SPELLER by Software International, STRIKE by S & K Technology, and AI-TYPIST by AIRUS Inc. To make it easier to refer to the implemented spelling program when making comparisons, it will be called by the name SCAN-SPELL.

PAPERBACK SPELLER is an interactive program which checks a user's document and provides a correction procedure for the misspelled words that it finds. It begins operation by searching the document for misspelled words. PAPERBACK SPELLER defines a word to be any combination of letters and the apostrophe with a length of 2 - 29, delimited by all other non word characters. The dictionary contains 60,000 words, including all letters of the alphabet, many proper names, names of countries and many cities, abbreviations, and some foreign words. The case of the letters is ignored. Words longer than twenty-nine letters are ignored (assumed to be correct). When PAPERBACK SPELLER finds a misspelled word, it displays and highlights the word to the user in context. Also displayed is a list of no more than five candidate corrections. These five candidate words seem to be the first five found in the dictionary and no attempt is made to find the most likely candidates. The user now has several options from which to choose. He may replace the misspelled word with one of the candidate corrections. He may replace the misspelled word with a correction of his choice which is not in the candidate list. He may ignore the misspelled word, leaving it as it is. Once the user chooses which action to take, the program resumes the search for the next word.

After the document has been completely read and corrected it is retained under its original file name. The original form of the document, before the changes, is put into a backup file. Statistics are given as to how many words were in the document, how many were misspelled, and how many were corrected.

PAPERBACK SPELLER tries to identify misspelled words which exhibit one or more of the four error types (substitution, deletion, insertion, or transformation). In addition, PAPERBACK SPELLER tries to detect words which run together, like 'andhe', and words which have been typed twice in a row, like 'and and he'. It does not make the assumption that misspelled words contain only one error but the correction rate is low for words which contain more than one error. Also, the spelling program depends heavily on the correctness of the first letter in the misspelled word. It does not correct words very efficiently when the first letter is actually wrong.

STRIKE is another interactive spelling program which both checks a document for misspelled words and provides the user with a list of candidate corrections when one is found. STRIKE, however, must be used in conjunction with a word processor. It is not a stand alone program. STRIKE checks the document for spelling errors page by page or paragraph by paragraph, but not the whole document at once. A word is defined by STRIKE to be a group of 1 to approximately 15 letters, including the apostrophe, separated by all other non word characters except digits. Words containing any digits are ignored (assumed to be correct). Some run on words can be detected. Letter case is also ignored. The dictionary used by STRIKE contains 49,000 words. After an entire block (page or paragraph) has been checked, the misspelled words are highlighted. STRIKE displays the document a screen at a time, not just a few lines, for context. The user may move the cursor to each highlighted misspelled word. He may choose a word from the candidate list to replace the word, he may choose to edit the word, or he may ignore it. After this, the user must specify that he wishes to check the next block of the text.

Since STRIKE operates through a word processor, the dispositions of the changed document and the original document are dealt with by this word processor and not by STRIKE. Also, all of the STRIKE commands are displayed in pop up menus in a corner of the screen. If the user wants to see a menu, he uses a function key to display it. He must use the key again to get rid of it before he can perform any more functions with STRIKE. STRIKE gives no statistics at the end of processing and is harder to use than PAPERBACK SPELLER.

.AI:TYPIST is the third commercial spelling program investigated. This is a simple word processing package with an option which can check spelling. A word is defined to be two or more letters delimited by blanks, any punctuation characters, or numerals. There is a main dictionary to which can be added other words. This dictionary contains 26,000 of the most often used words. The case of letters is ignored. The document appears on the screen with all the found possible misspellings marked with inverse video. These misspellings can be edited and the file saved with the correct changes at the end of the session. In this way, the spell checker is just a part of the editing process. There are no statistics. This package does not correct the words.

SCAN-SPELL, the program developed for this project, is a batch program. Instead of highlighting the misspelled words and displaying a list of candidate corrections, it produces a file which contains a list of misspelled words and a file which contains a list of candidate corrections. In addition, it produces another file which contains the document with as many misspelled words corrected as possible. The dictionary which SCAN-SPELL uses contains approximately 60,000 words. However, it does not contain any one letter words, any foreign words, nor any abbreviations.

From the descriptions of each of these four spelling programs, it is possible to predict a little about the behavior of each on a scanned document. PAPERBACK SPELLER and SCAN-SPELL have the most complete dictionaries and so should tag as incorrect a smaller group of words which are actually correct than AI:TYPIST and STRIKE. Strike ignores all words containing digits which means that it will not detect many of the misspelled words in the scanned document set because so many mistakenly contain digits. PAPERBACK SPELLER, STRIKE, AND AI:TYPIST will most likely read a word incorrectly if it contains any punctuation marks. On the other hand, SCAN-SPELL has been developed to properly read both words which contain digits and words which contain punctuation marks. Of the three commercial spelling programs, PAPERBACK SPELLER has the best performance. AI:TYPIST gives no list of possible corrections and STRIKE misses too many misspelled words. Also, the dictionaries of STRIKE and AI:TYPIST are less than adequate.

Formal tables have been calculated to compare PAPERBACK SPELLER and SCAN-SPELL although informal comparisons are done between all four of the spelling programs. The Tables 11, for scanned document characteristics, 12a - b and 13a - b, for the after spelling check analysis, 14a - b, for the after spelling correction analysis, and 15a - b, for the system summaries, parallel those tables which were done from the analysis of the pre-test documents.

The document scanned and used for the comparison of the four spelling programs was again taken from various school texts [ref. 17]. This document contains 6950 letters in 1629 words. The total number of words in this document and the distribution frequency of the letters is given in Table 10. Even though the words in the text are all simple, the most common words should be

Table 10. Word count and letter frequencies for document used for comparison tests.

# of words in document : 1629
# of words >1 in document : 1578

A	B	C	D	E	F	G	H	I	J	K	L	M
5	5	6	2	3	7	10	17	7	2	8	0	7

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6	8	2	0	4	15	25	1	0	7	0	1	0

a	b	c	d	e	f	g	h	i	j	k	l	m
620	77	139	317	824	180	183	454	425	1	60	293	153

n	o	p	q	r	s	t	u	v	w	x	y	z
462	507	101	4	444	418	589	152	56	219	1	122	1

Table 11. Data analysis of scanned document used for external comparisons of spelling programs SCAN-SPELL and PAPERBACK SPELLER.

document id	total no. words in document	# words in document with >1 letter		
		total	correct	incorrect
OCR A	1629	1578	1505	73
OCR B	1629	1578	1191	387
COURIER	1629	1578	797	781
ELITE	1629	1578	1184	394
GOTHIC	1629	1578	1393	185
totals	8145	7890	6070	1820

document id	incorrect					
	total	sub	ins	del	split	>1 err
OCR A	73	58	1	1	0	13
OCR B	387	317	1	1	10	58
COURIER	781	478	0	1	0	302
ELITE	394	231	2	1	0	160
GOTHIC	185	152	0	1	14	18
totals	1820	1236	4	5	24	551

Table legend:

sub = substitution error
ins = insertion error
del = deletion error
split = split word
>1 err = more than one error in word

Table 12a. SCAN-SPELL data analysis of correct words in scanned document used for external comparisons of spelling programs after spelling check.

document		correct	
id	total	tagged	not tagged
OCR A	1505	33	1472
OCR B	1191	28	1163
COURIER	797	15	782
ELITE	1184	31	1153
GOTHIC	1393	25	1368
totals	6070	132	5938

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 12b. PAPERBACK SPELLER data analysis of correct words in scanned document used for external comparisons of spelling programs after spelling check.

document id	total	correct tagged	correct not tagged
OCR A	1505	26	1479
OCR B	1191	24	1167
COURIER	797	9	788
ELITE	1184	24	1160
GOTHIC	1393	20	1373
totals	6070	103	5967

Table legend

tagged = correct word was mistakenly identified as a misspelling.

Table 13a. SCAN-SPELL data analysis of incorrect words in scanned document used for external comparisons of spelling programs after spelling check.

document	incorrect								
	sub			ins			del		
id	tot	yes	no	tot	yes	no	tot	yes	no
OCR A	58	56	2	1	1	0	1	1	0
OCR B	317	297	20	1	1	0	1	1	0
COURIER	478	468	10	0	0	0	1	1	0
ELITE	231	224	7	2	2	0	1	1	0
GOTHIC	152	120	32	0	0	0	1	1	0
totals	1236	1165	71	4	4	0	5	5	0

document	incorrect					
	split			>1 err		
id	tot	yes	no	tot	yes	no
OCR A	0	0	0	13	13	0
OCR B	10	10	0	58	57	1
COURIER	0	0	0	302	297	5
ELITE	0	0	0	160	159	1
GOTHIC	14	11	3	18	15	3
totals	24	21	3	551	541	10

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

Table 13b. PAPERBACK SPELLER data analysis of incorrect words in scanned document used for external comparisons of spelling programs after spelling check.

document					incorrect				
	sub				ins			del	
id	tot	yes	no	tot	yes	no	tot	yes	no
OCR A	58	52	6	1	1	0	1	1	0
OCR B	317	255	62	1	1	0	1	1	0
COURIER	478	301	177	0	0	0	1	1	0
ELITE	231	182	49	2	2	0	1	1	0
GOTHIC	152	63	89	0	0	0	1	1	0
totals	1236	853	383	4	4	0	5	5	0

document		incorrect				
		split			>1 err	
id	tot	yes	no	tot	yes	no
OCR A	0	0	0	13	13	0
OCR B	10	9	1	58	55	3
COURIER	0	0	0	302	216	86
ELITE	0	0	0	160	115	45
GOTHIC	14	11	3	18	10	8
totals	24	20	4	551	409	142

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

yes = misspelled word is correctly flagged as a misspelled word.
no = misspelled word is not flagged.

Table 14a. SCAN-SPELL data analysis of scanned document after correction used for external comparisons of spelling programs.

document id	sub			ins			del		
	tot	cor	not	tot	cor	not	tot	cor	not
OCR A	56	48	8	1	0	1	1	0	1
OCR B	297	285	12	1	0	1	1	0	1
COURIER	468	247	221	0	0	0	1	0	1
ELITE	224	197	27	2	0	2	1	0	1
GOTHIC	120	105	15	0	0	0	1	0	1
totals	1165	882	283	4	0	4	5	0	5

document id	split			>1 err		
	tot	cor	not	tot	cor	not
OCR A	0	0	0	13	0	13
OCR B	10	0	10	57	0	57
COURIER	0	0	0	297	0	297
ELITE	0	0	0	159	0	159
GOTHIC	11	0	11	15	0	15
totals	21	0	21	541	0	541

Table legend:

sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word

cor = misspelled words had true correction in candidate list

not = misspelled words had no true word in the correction list.

Table 14b. PAPERBACK SPELLER data analysis of scanned document after correction used for external comparisons of spelling programs.

document id	sub			ins			del		
	tot	cor	not	tot	cor	not	tot	cor	not
OCR A	52	43	9	1	1	0	1	1	0
OCR B	255	175	80	1	1	0	1	1	0
COURIER	301	97	204	0	0	0	1	1	0
ELITE	182	80	102	2	2	0	1	1	0
GOTHIC	63	13	50	0	0	0	1	1	0
totals	853	408	445	4	4	0	5	5	0

document id	split			>1 err		
	tot	cor	not	tot	cor	not
OCR A	0	0	0	13	3	10
OCR B	9	0	9	55	3	52
COURIER	0	0	0	216	8	208
ELITE	0	0	0	115	23	92
GOTHIC	11	4	7	10	1	9
totals	20	4	16	409	38	371

Table legend:
sub = substitution error.
ins = insertion error.
del = deletion error.
split = split word.
>1 err = more than one error in word.

cor = misspelled words had true correction in candidate list.
not = misspelled words had no true word in the correction list.

Table 15a. System summary for SCAN-SPELL.

		scanned document	corrected* document	
document id	total # words w/ >1 letters	# words correct	# words correct	system efficiency
OCR A	1578	1505	1553	66%
OCR B	1578	1191	1476	74%
COURIER	1578	797	1044	32%
ELITE	1578	1184	1381	50%
GOTHIC	1578	1393	1498	57%
totals	7890	6070	6952	47%

* after automatic spelling correction by the program.

Table 15b. System summary for PAPERBACK SPELLER.

document id	total # words w/ >1* letters	scanned document	corrected document	system efficiency
		# words correct	# words correct.	
OCR A	1578	1505	1572	92%
OCR B	1578	1191	1512	83%
COURIER	1578	797	1315	66%
ELITE	1578	1184	1484	76%
GOTHIC	1578	1393	1478	46%
totals	7890	6070	7363	71%

*after user has corrected all words which were found to be incorrect by the program.

the same as those in everyday English. These texts simply lack the inclusion of complicated, longer words which, if correctable, have more of a chance of being properly corrected in the final output document.

As shown in Table 11, the rates of misspelled words for each font is 5% for OCR A, 25% for OCR B, 49% for COURIER, 25% for ELITE, and 12% for GOTHIC. Once again, the data shows that OCR A and GOTHIC have the best scanner performance and COURIER the worst. Also, substitution errors make up the bulk of the mistakes (68 per cent), with multiple errors then split words coming next (30 per cent and 1 per cent). Insertion and deletion errors continue to be almost nonexistent.

Tables 12a and 12b show the disposition of the correct words during the spelling check for each of the spelling packages SCAN-SPELL and PAPERBACK SPELLER. SCAN-SPELL seems to mark as incorrect more correct words than PAPERBACK SPELLER; i.e. SCAN-SPELL marks 2 percent and PAPERBACK SPELLER marks 1.5 per cent. This is not necessarily a big problem in the long run. PAPERBACK SPELLER has abbreviations in its dictionary while SCAN-SPELL does not. This means that SCAN-SPELL will mark as incorrect all abbreviations occurring in the text, even when they are properly spelled. However, because PAPERBACK SPELLER allows these abbreviations in the text, many short misspelled words are ignored when they have the appearance of a correctly spelled abbreviation. This is a more serious problem, resulting in many incorrect words not detected and thus not corrected. As shown in tables 13a-b, SCAN-SPELL does find 14 per cent more incorrect words than PAPERBACK SPELLER (95 per cent verses 71 per cent).

STRIKE and AI:TYPIST both mark many more correct words as incorrect than either PAPERBACK SPELLER or SCAN-SPELL. Examples of words marked by STRIKE include 'swan', 'stairway', 'Canada', 'steamship', and 'David'. Examples of words marked by AI:TYPIST include 'twinkled', 'monkeys', 'amongst', and 'leopard'. This is really just a dictionary problem, however, and these words can be added to both the dictionary for STRIKE and the one for AI:TYPIST.

In the analysis of the incorrect words which both spelling programs found, it is seen that SCAN-SPELL found 14 per cent more incorrect words than PAPERBACK SPELLER (95 percent verses 71 per cent). One problem with PAPERBACK SPELLER is, as stated, that it has so many abbreviations in its dictionary that short misspelled words are not detected when they are transformed into one of these abbreviations. Another problem with the detection rate of PAPERBACK SPELLER is that it misses more split words and some words where a letter has been mistaken as a punctuation mark. In this case, when there are punctuation marks in a word or a part of a split word, PAPERBACK SPELLER will not include them when it reads the word. Therefore, the resulting PAPERBACK SPELLER version of the word may be too short, or it may be found to be correct. An example of this would be the word 'studio' misspelled as 'stud!o'. PAPERBACK SPELLER reads this word as 'stud', ignores the '!' because it is a punctuation mark, and ignores the 's' because it is only one letter. Thus, 'stud!o' is found to be a correct word and is not tagged by PAPERBACK SPELLER. Another type of misspelling which PAPERBACK SPELLER will not detect is a misspelling due to the wrong case of the letter, because PAPERBACK SPELLER ignores case when reading words. AI:TYPIST and STRIKE have the same general problems as PAPERBACK SPELLER in detecting these kinds of misspellings. In addition, STRIKE ignores all words which have any digits in them. This is a serious disadvantage when dealing with text produced from optical scanner software as many words have digits in them.

The tables which show the analysis of incorrect words which are corrected (14 a - b) are a little different from the ones used for the pre-tests. The column labelled 'cor' gives the count of all misspelled words which had the proper correction in their correction list. The column labelled 'not' gives the count of all misspelled words which had no proper candidate correction. From these two variables, it is seen that SCAN-SPELL has a higher rate of candidate corrections than PAPERBACK SPELLER, 51 per cent verses 36 per cent. One reason for this is that PAPERBACK SPELLER relies heavily on the correctness of the first letter in the misspelled word. If this letter is misspelled, PAPERBACK SPELLER rarely gives the proper correction in the candidate list. However, it should be noted that the correction rate for a text corrected by PAPERBACK SPELLER can be 100 per cent if, for every misspelled word, the user is able to indicate the correct spelling to the program. The user can do this in one of two ways, by choosing a word from the candidate list, or by typing in the correction himself. Since choosing the word from a candidate list is much easier, this statistic was collected in the comparison between the spelling programs for tables 14a and 14b, under the column labelled 'cor'.

It is difficult to compare the programs on corrected text because, if used in an efficient way by the user PAPERBACK SPELLER may have a near 100 per cent correction rate. For this reason, the system summary for PAPERBACK SPELLER, table 15b, is misleading. The table assumes that all words which could have been corrected, would be corrected by the user. This is in opposition to the system summary for SCAN-SPELL which counts only the words properly corrected by the automatic correction program. Although the assumption of an infallible user leads to perfect correction of misspelled words in PAPERBACK SPELLER, the misspelled words which PAPERBACK SPELLER does not detect can not be corrected. In fact, PAPERBACK SPELLER did not do well in the correction rate for GOTHIC (it performed less well than SCAN-SPELL with 46 verses 57 per cent), and only the correction rate for OCR A (92 per cent) is very near 100 per cent. The problem is that PAPERBACK SPELLER does not detect some types of misspelled words and certain fonts are adversely affected. These types include short misspelled words which are taken to be abbreviations or other correct words by the PAPERBACK SPELLER dictionary and misspelled words which contain punctuation marks or numerals, thus effectively dividing them into shorter words. The system summary table for PAPERBACK SPELLER does point out one of its advantages however, mainly that it can have a very high correction rate. This presumes that the user is industrious and infallible, that most misspelled words have been detected, and that an appropriate font has been used.

The system summary for SCAN-SPELL indicates that the corrected text which it produces has a good rate of correction for some fonts. OCR B (74 per cent) has the best rate, but OCR A (66 per cent) and GOTHIC (57 per cent) also have good rates. Since these correction rates are at least adequate, they indicate that statistics taken from the sample are good for other documents as well. These rates indicate that the method of automatically producing a corrected text can be a viable part of a spelling program when used for an optical scanner.

A major difference in the behavior of PAPERBACK SPELLER and SCAN-SPELL is due to the differences in behavior between an interactive spelling program and one which works in batch mode. A spelling program which operates interactively allows certain document manipulation to take place. The spelling processes can be monitored, and corrections always done to any words which the program deems as misspelled. If the spelling program has been adapted to the user's environment and has a high rate of correctly identifying misspelled words, a corrected version of the text will be nearly perfect. This perfection is, however, not easily obtained. The user must make all the corrections manually, as they are found by the program. If the document is a long one, this is a tedious, labor intensive, and therefore expensive process. With a program which runs in batch mode, the system does not need to be monitored and if the spelling process is a long one, at least there are no man hours needed to oversee it. If the correction rates of both systems are nearly equal then, it would seem that the batch system would be preferable. However, if the interactive system gives better results, then it would depend on how big the difference between the results was, and also on the convenience of the user. Finally, the actual time each system takes may be a factor. A batch program can generally take a longer period of time than an interactive one because it runs alone and needs no user maintenance. However, if it takes too long, it will still be a problem for the user, who may need the facilities for other functions. The SCAN-SPELL spelling program does take a long time to run. However, the documents used for the tests were long ones, probably not of a normal length. It would take any spelling program a long time to check and correct these documents. It was hard to measure the actual length of time it took PAPERBACK SPELLER to process the document because this time would necessarily have to include the user response time as well. However, in a very loose measurement, it seems that it takes SCAN-SPELL about twice to three times as long as PAPERBACK SPELLER.

10. DISCUSSION

The design ideas with which SCAN-SPELL was shaped may be quite helpful when attempting to correct documents produced from scanner software. But in terms of functions and ease of use, the spelling program SCAN-SPELL does not seem to be as complete a program as PAPERBACK SPELLER. There are several reasons for this and they suggest ways in which the system can be improved.

A feature which all three commercial programs offered which is not a part of SCAN-SPELL is the existence of an auxiliary dictionary. It is important that a dictionary not be too big and it is impossible that a dictionary could ever incorporate all the words which every user may wish to employ in their documents. An auxiliary user's dictionary can be established to deal with this problem. Uncommon words which appear in a document but not in the program's dictionary can be added to this user's dictionary. It can usually be manipulated in a way which the main dictionary cannot, so words can be inserted and deleted to match any user profile. This is a good way to deal with the specialized language which goes with certain technical areas. Such a dictionary is not a part of SCAN-SPELL.

If an interactive mode were added to the SCAN-SPELL program, it would gain some of the advantages of the other type of systems. A user would then have a choice of mode, something that does not exist in any commercial systems. An interactive mode would be useful for a short document which did not have many errors, and those it did have were simple substitutions.

Other system improvements to the SCAN-SPELL program would include a better user interface, a reduction in operating time, and better correction algorithms. Both groups of split words and multiple errors could at least be partially corrected.

Automatic correction of misspelled words using probabilistic analysis is a reasonable way to produce a corrected text as the final output of optical scanner software. The advantage of this system is that there is no operator intervention.

The efficiency of the correction algorithms depends not only on the soundness of their design but also on the analysis and appropriateness of the training documents. No probabilistic system can be guaranteed to be 100 per cent efficient because the act of making a choice based on probability implies the possibility of making a wrong choice. The document must have a very high percentage of the misspelled words corrected to satisfy most users. If correction must be done automatically, context must play some role in the selection of the candidate words. There are several ways in which this context or meaning may play a role.

A program which did simple grammar checking would aid in the proper selection of a correct word from the list of candidate corrections. Only words which were grammatically correct would be considered. Ideally, the more complicated the grammatical analysis, the more likely it would be to determine the correct spelling of the word. This type of checking is, however, a complicated procedure and would undoubtedly add to the time it took to process a document.

Another kind of context could also be used to find corrections to the misspelled words. If the scanner software could keep track of alternative choices for the characters it reads, these choices could be used in forming candidates for the misspelled word. For instance, if the scanner software chooses '!' as one of the characters it reads, but had as another possible choice (which was then rejected) the letter 'l', then this information could be given to the spelling program. The spelling program could then use the information in forming candidate corrections to misspelled words rather than relying on the long-term probability.

In the end, spelling checkers cannot possibly find all the mistakes which occur in a document file. Words which are split (which have an insertion error of a space), one letter words which are wrong, numbers and punctuation characters which are wrong or which have been mistaken for the letters of a word, words which are wrong but which also occur in the dictionary, and words where an upper case character has been mistaken for its lower case equivalent are all major problems which make the identification of mistakes difficult.

It is important then, to consider applications where the performance of an ideal spelling program for an optical scanner can be of use. First of all, the characteristics of an ideal spelling program should be given. This is one which is easy to use, takes as little user intervention as possible, and corrects as many misspelled words as possible. Not all words would be corrected, so applications which needed such perfection would not be considered. However, memos, notes, and first draft reports and papers could be stored on the computer using a system involving an optical scanner and a spelling program. In fact, any information which is not needed for some kind of formal presentation can be stored using this kind of system.

The advantages of the SCAN-SPELL spelling program are the unique customization of the spelling program to the scanner software and the idea of automatic spelling correction. But using SCAN-SPELL means accepting less than perfection in the trial document. In all likelihood, probabilistic methods such as SCAN-SPELL cannot fix all errors. The approaches to improve the system include using context to exclude candidate corrections and using information from the scanning software to identify the likely errors in the scanned document and the most likely corrections to those errors. Also, of course, improvements in the scanning algorithms could reduce the number of words needing correction.

11. APPENDICES

11.1 APPENDIX A

OCR A

TRUE LETTERS

			a	b	c	d	e	f	g	h	i	j	k	l	m
L	B	a	817												
E	Y	b		158											
T		c			257										
T	S	d				365									
E	C	e					1298								
R	A	f						192		2					
S	N	g							226						
·	N	h								611					
R	E	i									505				
E	R	j										21			
P		k											155		
O		l				1					26			451	
R		m													250
T		n								19					
E		o		4		12			17						
D		p													
		q							20						
		r													
		s													
		t													
		u													
		v													
		w													
		x									8		1		
		y													
		z									9			1	
		D		1											
		J										3			
		&											2		
		’						2			1				
		\				8									
		+												2	
		‘												1	
		!										1			

OCR A - cont.   TRUE LETTERS

LETTERS REPORTED BY SCANNER	n	o	p	q	r	s	t	u	v	w	x	y	z
a													
b													
c							1						
d													
e													
f													
g													
h													
i													
j													
k													
l													
m													
n	619												
o		773	26				2						
p			143										
q				18									
r					621								
s						650	1						
t							832						
u								255	2				
v									95			20	
w										282			
x											32		
y												169	
z							5						15

OCR A - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	38								1																	
B		15																								
C			19																							
D				21																						
E					15																					
F						13																				
G							12																			
H								39																		
I									42																	
J										11																
K											12															
L												30														
M													23													
N														19												
O															21											
P																24										
Q																	14									
R																		16								
S																			32							
T																				67						
U																					10					
V																						10				
W																							28			
X																								10		
Y																									14	
Z																										10
f													2													

OCR B TRUE LETTERS

LETTERS REPORTED BY SCANNER

	a	b	c	d	e	f	g	h	i	j	k	l	m
a	798												
b		151											
c			257		178								
d				355									
e					1076								
f						138							
g							258						
h								616					
i									451	2			
j										20			
k											151		
l												450	
m													250
n								16					
o		1		21			4						
p							1						
q													
r													
s	18				44								
t						18							
u													
v													
w													
x											1		
y													
z													
D		11											
I						4							
K											5		
O				4									
Q				4									
0				1									
1									65				
(						31							
:												1	
"	1												
,									26			4	
*											1		
!									7				
/						3							
)										3			
#				1									

OCR B - cont. - TRUE LETTERS

LETTERS REPORTED BY SCANNER

	n	o	p	q	r	s	t	u	v	w	x	y	z
a													
b													
c							20						
d													
e													
f													
g				3									
h													
i													
j													
k													
l							32						
m													
n	619												
o		762	3										
p		11	107										
q				15									
r					621								
s						650							
t							737						
u								255					
v									89				
w									8	57			
x											32	1	
y												166	
z							8						15
C							4						
D			40										
I							10						
P			19										
X												21	
Y												1	
1							15						
:							2						
(							13						

OCR B - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	38																									
B		15																								
C			19																							
D				21											6											
E					15																					
F						13																				
G							12																			
H								39						6								6				
I									42																	
J										11																
K											12															
L												30														
M													25	1												
N														12												
O															14											
P																24										
Q															1		14									
R																		16								
S																			27							
T																				27						
U																					10					
V																						9				
W																							22			
X																								10		
Y																									14	
Z																										10
s																			1							
v																						1				
5																			4							
*									1																	

COURIER TRUE LETTERS

LETTERS REPORTED BY SCANNER

	a	b	c	d	e	f	g	h	i	j	k	l	m
a	679												
b		160						68			2		
c			234										
d				375									
e					1286								
f						193							
g							232						
h								557					
i									419				
j										22			
k											154		
l									33			269	
m													249
n		1						6					
o		2	5	11			15						
p													
q							2						
r													
s	92		3		12								
t													
u								1			1		
v													
w													
x											1		
y													
z	46		15			1							
1									97			182	
3										3			
9							13						
*												4	
^							1						
"													1

COURIER - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER	n	o	p	q	r	s	t	u	v	w	x	y	z
a													
b							22						
c			1				1						
d													
e													
f													
g				1									
h													
i													
j													
k													
l													
m													
n	561				3			26					
o		771	18	2									
p			141										
q				15									
r	1				618		12						
s						499							
t							801						
u	55							229		1			
v									96			7	
w										278			
x										1	32		
y												178	
z		2				151							15
D			1										
F			2										
I							1						
P			6										
Y												4	
&							4						
,									1	1			
"	2												

COURIER - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	35																									
B		14			1																					
C			19																							
D				10												1										
E					14																					
F						13										18										
G							12																			
H								39					1	6									2			
I									42																	
J										11																
K											12															
L												30														
M													23	5												
N														8									14			
O				10											20						2					
P																5										
Q																	14									
R																		16								
S																			23							
T																				67						
U																					8					
V																						9				
W																							12			
X																								10		
Y																						1			14	
Z																										8
o															1											
s																			7							
z																										2
0				1																						
1									1																	
8		1																	2							
!													1													
&	2																									

ELITE

TRUE LETTERS

LETTERS REPORTED BY SCANNER

	a	b	c	d	e	f	g	h	i	j	k	l	m
a	811							1					2
b		158		1				16					1
c			255	2									
d				371			1						
e			1		1286		1						
f						190							
g							241						
h								537					1
i									521				
j										21			
k											156		
l								5				266	
m	1				7								232
n				2				8					1
o		4		6									10
p													
q		1					1						
r						2			11			3	
s													1
t						1							
u													
v													
w													
x	1		1		2								
y													
z	3				1		1						
B							4				1		
E	1						11						
I				1				11	7			21	
K											1		
M				2			1		1				
R									1				
S									1				
T									1				
V								1					
Y						1			2				
1								30				162	
3									1	3			
7									1				
?								11					
[								5					
!									2				
"													1
*												1	
]										1			

ELITE - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	n	o	p	q	r	s	t	u	v	w	x	y	z
a						3							
b													
c													
d													
e						7							
f													
g													
h										1			
i													
j													
k										3			
l	5												
m	2	3			1			4		32			
n	565	1	2					1					
o	3	767	9										
p			158										
q				18									
r	27				618		4	1			1	1	
s						632						6	
t							825						
u								247		51			
v								1	96	1		8	
w										187			
x	1	1			1		1				21		
y												161	
z	2					5					10		10
E							1						
F												3	
K												1	
M										1			
U								1					
V												3	
W										2			
Y												11	
3												3	
?	1												
'	1												
@							1						
"										1			
<							1						

ELITE - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	38																									
B		15																								
C			18				1																			
D				21																						
E					15																					
F						13																				
G			1				11																			
H								39																		
I									42																	
J										11																
K											12															
L												29														
M													25													
N														19									1			
O															21											
P																24										
Q																	14									
R																		16								
S																			32							
T																				67					1	
U																					10					
V																						10				
W																							27			
X																								10		
Y																									13	
Z																										10

GOTHIC

TRUE LETTERS

LETTERS REPORTED

BY SCANNER

	a	b	c	d	e	f	g	h	i	j	k	l	m
a	817				4								
b		162											
c			257										
d				378									
e					1294								
f						157							
g							242						
h								631					
i									516				
j										24			
k											157		
l												454	
m													250
n								1					
o				1			12						
p													
q													
r													
s													
t						5							
u		1					2						
v													
w													
x													
y													
z													
B							1						
F						32							
I									1				
J										1			
0				7									
9							6						
&											1		
!									30			1	
'									2				

GOTHIC - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	n	o	p	q	r	s	t	u	v	w	x	y	z
a													
b													
c							1						
d													
e													
f													
g													
h													
i													
j													
k													
l													
m													
n	619									1			
o		773	12										
p			136										
q				17									
r					621								
s						650	3						
t							806						
u								255					
v									97			5	
w										281			
x											32		
y												183	
z													15
I							4						
L							12						
P			20										
Y												1	
[							14						
&							2						

GOTHIC - cont. TRUE LETTERS

LETTERS REPORTED BY SCANNER

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	38																									
B		15		1														3								
C		19																								
D				20											7											
E					15																					
F						13																				
G							4																			
H								39																		
I									42																	
J										11																
K											12															
L												30														
M													20													
N													4	19												
O															7											
P																24										
Q																	14									
R																		16								
S																			18							
T																				67						
U																					10					
V																						9				
W													1										28			
X																								10		
Y																						1			14	
Z																										10
0															7											
5																			14							
6							8																			
!									1																	

11.2 APPENDIX B

A USER'S GUIDE

There are three options to choose from in this spelling package. Option a checks the document for spelling mistakes. Option c corrects the BADWRDS file (output from the checker). Option b does both at once. The spelling checker can be used by typing

```
CHECKER filename for all options
```

where filename is the name of the file to be checked for possible misspellings. Options b and c will ask for a font identifier. Knowing the proper font for the file is very important to the correction algorithms. Possible fonts are OCR A, OCR B, COURIER, ELITE, and GOTHIC. Option a and the checker part of option b use as input the user specified file. They output two files, a file for the user called WRONG.WRD which gives a list of all possible misspellings with the line numbers where they occur, and a file to be used by the correction program called BADWRDS which simply lists all the possible misspellings. Option c and the correction part of option b use as input the file BADWRDS. They output a file called CAND.WRD which has all possible misspellings with their possible corrections if any. In order to use the package correctly, the files CHECKER.EXE and SDICT.DIC must be present in your directory.

12. BIBLIOGRAPHY

[1] Srihari, S., N., Computer Text Recognition and Error Correction, IEEE Computer Society Press, New York, 1985.

[2] Dunn, Eric, "Dictionary Compression and Decomposition", Byte, vol. 9, no. 10, pp. 457-459, 1984.

[3] McIlroy, M., D., "Development of a Spelling List", IEEE Transactions on Communications, vol. COM-30, no. 1, pp. 91-99, 1982.

[4] Peterson, J., "Computer Programs for Detecting and Correcting Spelling Errors", Communications of the ACM, vol. 23, no. 12, pp. 676-687, 1980.

[5] Peterson, J., Design of a Spelling Program: an experiment in program design, Springer-Verlag, New York, 1980.

[6] Pollack, J., J., and Zamora, A., "System Design for Detection and Correction of Spelling Errors in Scientific and Scholarly Text", JASIS, vol. 35, no. 2, pp. 104-109, 1984.

[7] Zamora, A., "Automatic Detection and Correction of Spelling Errors in a Large Database", JASIS, vol. 31, no. 1, pp. 51-57, 1980.

[8] Turba, T., N., "Checking for Spelling and Typographical Errors in Computer-Based Text", SIGPLAN Notices, vol. 16, no. 6, pp. 51-60, 1981.

[9] Liang, F., M., "Word Hy-phen-a-tion by Computer", Dept. of Comp. Sci., Stanford Univ., Report no. STAN-CS-83-977, 1983.

[10] Carter, L., Floyd, R., Gill, J., Markowsky, G., Wegman, M., "Exact and Approximate Membership Testers", Proceedings of the Tenth Annual ACM Symposium on the Theory of Computing, pp. 5965, 1978.

[11] "AI:TYPIST Manual", AIRUS Inc., 1986.

[12] "PAPERBACK SPELLER", Paperback Software Int., 1985.

[13] "STRIKE", S & K Technology Inc., 1986.

[14] Freund, John, E., Mathematical Statistics, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.

[15] Kucera H., Francis, W. N., Computational Analysis of Present Day American English, Brown University Press, Providence, RI, 1967.

[16] "IEEE Recommended Practice for Speech Quality Measurements", IEEE Transactions on Audio and Electoacoustics, vol. AU-17, no.3, pp. 225-246, 1969.

[17] C. Y. Suen, personal communication.