

Post-Editing through Approximation and Global Correction

Kazem Taghva*, Julie Borsack, Bryan Bullard, and Allen Condit
Information Science Research Institute
University of Nevada, Las Vegas

March 1993

Abstract

This paper describes a new automatic spelling correction program to deal with OCR generated errors. The method used here is based on three principles:

1. Approximate string matching between the misspellings and the terms occurring in the database as opposed to the entire dictionary
2. Local information obtained from the individual documents
3. The use of a confusion matrix, which contains information inherently specific to the nature of errors caused by the particular OCR device

This system is then utilized to process approximately 10,000 pages of OCR generated documents. Among the misspellings discovered by this algorithm, about 87% were corrected.

*Email: taghva@cs.unlv.edu.

1 Introduction

Optical character recognition (OCR) devices are currently used to convert printed materials into ASCII text for automated information retrieval. The process generally consists of scanning the document to produce images, converting the images to ASCII text, followed by manually correcting errors caused by OCR. The last step is by far the most expensive part of the process. At first, it seems that the following strategies partially resolve the problem:

1. Employing a spell-checker to identify and correct misspellings (possibly semiautomatically), and
2. Allowing the text retrieval system to use some sort of approximate matching to find syntactically close hits.

The first approach does not work well because spell checkers are mainly built to correct errors that are caused by either the conventional keyboard arrangement (i.e. a may be mistyped as s) or to correct common spelling errors caused by doubling, undoubling, or transposition of characters. Furthermore, the literature lacks studies related to the nature of errors caused by OCR. For example, Pollock and Zamora state that “. . . *OCR output contains almost exclusively substitution errors. . .*”[6]. Our experiment with OCR indicates deletion, insertion, and substitution of a sequence of characters for one character (or vice versa) are common[8].

The second approach works well either with a very small database, or if the text is fairly clean. The database size and percentage of misspellings can lead to a considerable number of syntactically close hits. According to [7] the best OCR device on good quality input has 99% character accuracy, which could produce up to 25 misspelled words per page. Since manual correction is very expensive, the following questions arise:

1. What should be the character accuracy of an OCR device in order to avoid manual correction?
2. Can the correction be done automatically?

Experiments described in [12], which investigate the effects of OCR errors on text retrieval systems, are our attempt to answer the first question. The second question addresses the same problems spell-checking programs do. Our approach relies on approximation with respect to the words in the database as opposed to a standard dictionary. Our method is based on the assumption that for every misspelled word in the database, there are correctly spelled occurrences of the same word also in the database. Together with the information on the list of words in individual documents, and optical characteristics of errors captured in the confusion matrix, we are able to identify unique candidates to correct misspellings.

2 Preliminaries

Our post-processing system is designed to take advantage of the similarity of objects in the database. In particular, correctly spelled words with high frequency will be used to rectify misspellings. The source of errors in OCR generated text is different from errors in manually typed text[5]. The former is due to similarities of the character shapes as opposed to errors

caused by conventional keyboard arrangement in the latter. In OCR generated text, it is very likely that **i** be mistaken for either **l** or **1**, the character **c** may be read as **e**, and the uppercase character **N** incorrectly recognized as **IV**. In conventionally keyed-in documents, **a** may be mistyped as **s**. Our system will account for errors in the OCR text, especially optical errors, by effectively using an automatically generated *confusion matrix*.

Following Wagner and Fisher[14], let Σ be a finite alphabet. An *edit operation* is a pair of strings $(a, b) \neq (\epsilon, \epsilon)$ over Σ of length less than or equal to 1. We say string x results from w , in notation, $w \implies x$, if there are strings α and β such that $x = \alpha b \beta$ and $w = \alpha a \beta$. We say (a, b) is a *substitution* if $a \neq \epsilon$ and $b \neq \epsilon$, a *delete* operation if $b = \epsilon$ and $a \neq \epsilon$, and an *insertion* operation if $a = \epsilon$ and $b \neq \epsilon$. In the rest of the paper, we use the term *edit distance* n to refer to some combination of n applications of substitution, insertion, or deletion.

Among known approaches to error correction, there are three popular techniques as described in [5], [6], and [1]. The first technique employed in SPELL[5] generates a set of words one edit-distance away from the misspellings and then this set is checked against the dictionary to eliminate words which are not present in the dictionary. The remaining words are possible candidates for correcting the misspellings. Since the technique is dependent on the entire dictionary and pays no attention to optical errors—it was not designed for OCR error correction—in general it cannot identify a single candidate and thus relies on a decision from the user.

The second program, SPEEDCOP[6], is similar to hashing techniques. For each word in the dictionary, it builds a *skeleton key*¹ made up of the following characters: the first letter, the unique consonants in order of occurrence, and the unique vowels in order of occurrence. For example, the skeleton key for the word **information** would be **infrmtoa**. Upon discovering a misspelling, the skeleton key is found for it and is then tested against the skeleton keys in the dictionary to find possible candidates. Pollock and Zamora[6] argue that the key is chosen based on the following four principles:

1. The first letter is likely to be keyed-in correctly.
2. Consonants should be given more weight than vowels.
3. The original consonant order is mostly preserved.
4. The key is not altered by typical doubling or undoubling.

It is just as likely that an OCR device will incorrectly translate the first character as any other character in a word. Recognition devices will substitute vowels such as **e** and **i** for the consonants **c** and **l**, respectively. Furthermore, as stated previously, the use of the entire dictionary will lead to a sizeable list of possible candidates for each misspelling.

The third technique is trigram analysis[1]. In this approach, words in the dictionary are first converted to an inverted list consisting of trigrams. For example, the word **information** is broken into these units: **#in, inf, nfo, for, orm, rma, mat, ati, tio, ion, on#**. The inverted file keeps track of the list of the trigrams and words containing each trigram. Upon discovering a misspelling, the list of trigrams in the misspelling is used to identify the possible candidates for the misspelling. Trigram analysis works well for words with longer lengths, unless the dictionary is not rich enough to contain the appropriate candidates. Due to the fact that trigram analysis is flexible and extensible, it is a good choice for further study. The two shortcomings it has in relation to our work are:

¹Another key known as an *omission key* is also used to avoid problems associated with omission.

1. It uses the entire dictionary and pays no attention to the words locally available in the document.
2. It does not take advantage of the information pertinent to OCR errors—in particular, alphabetic characters which are read as punctuation marks will not be detected.

In [13], a method is described that in principle is based on the trigram technique. It selects candidate keys according to the frequency of characters in the dictionary. It picks the four rarest characters in the word, then makes all possible three-combinations of these four characters. This method, like trigram analysis, forms an inverted file for all the words in the dictionary and tests the misspelling against this list with the help of approximate matching to choose possible candidates. Again, this method pays no attention to the information available in each document and does not take advantage of any confusion matrix.

3 Global Editing

Our approach to error correction is oriented toward a text retrieval application. It is similar to the second pass of a two-pass assembler. We concentrate on a subset of important words in the collection: words indexed by the retrieval system (i.e. *non stop words*) with high frequency². We resolve ambiguities by using the local information available in each individual document and a *confusion matrix*. Although the confusion matrix has been used in OCR correction previously[11], our post-processing system builds the matrix it uses as it processes longer words. In what follows, we will describe three experiments with the third one representing the results of the complete post-processing system. In the first experiment, we loaded approximately 10,000 pages (150,000 distinct terms with 2,000,000 occurrences) of OCR generated documents into an inverted file text retrieval system, creating an index of non stop words. These pages were automatically zoned and processed; thus no manual processing was involved. Each word in the index was spell-checked. If it was a correctly spelled word and its frequency was high, it was added to the list of centroids. All other words were considered misspellings. As a result, the index of non stop words was divided into two sets:

Centroids— those words that were correctly spelled with respect to our dictionary³ and which had high frequency.

Misspellings— Centroids complement.

The next step was to cluster misspellings according to some measure of similarity as described in [3, 4, 10, 14]. We chose *agrep*[15] as our mechanism for cluster formation. *Agrep* or *approximate grep* can identify strings that differ from a given pattern by a specified edit distance. For each centroid, we run *agrep* against the set of misspellings. The basic assumption here is that the words in each cluster are similar to the centroid and thus should be equated to that centroid term.⁴ We manually examined each cluster to determine if the correct centroid was selected. The results are broken down in the tables by word length to emphasize the fact that as misspellings become shorter, the centroid selection becomes less certain and the proportion of correctly clustered misspellings decreases. Table 1 lists

²For this project, “high frequency” means words which occurred more than once in the collection.

³Our dictionary was the standard dictionary provided by *ispell*[2].

⁴For this paper, we are using *term* and *word* interchangeably.

word length through 18	centroids	populated centroids
18	1	0
17	5	2
16	8	1
15	23	2
14	75	23
13	200	62
12	339	120
11	639	223
10	946	372
09	1339	551
08	1574	753
07	1766	1060

Table 1: Centroids identifying misspellings in experiment 1

word length through 18	misspellings	occurrences of misspellings	incorrect replacements	occurrences of incorrect replacements
18	0	0	0	0
17	2	2	1	1
16	7	8	1	1
15	9	10	1	1
14	57	105	3	26
13	183	257	7	32
12	435	677	19	87
11	959	1461	62	155
10	1796	2629	123	321
09	3124	4933	251	708
08	5134	9050	460	1524
07	8036	16367	995	3684

Table 2: Incorrect replacements of misspellings in experiment 1

the total number of centroids at each word length and in the third column, the number of centroids that clustered misspellings. Table 2 shows the cumulative total of misspelled terms, their number of occurrences in the text,⁵ the number of terms that would be erroneously changed if the misspellings were equated to their centroid, and lastly the number of erroneously changed occurrences. Since in this experiment we had no method for resolving ambiguity for multiply clustered misspellings, these terms were counted as errors.

In our first run, a number of our “misspellings” were actually correctly spelled terms. For example, **preconstruction** was considered a misspelling—it was clustered with **reconstruction**. Furthermore, the assumption that all low frequency terms are misspellings was also wrong. So using the same document collection, our second experiment addresses these problems.

Since our documents are specific to the Licensing Support System (LSS)[9], we decided to augment our dictionary with terms from this specialized domain. This augmentation

⁵A misspelled term may appear more than once.

word length through 18	centroids	populated centroids
18	86	0
17	103	1
16	185	1
15	262	3
14	340	24
13	568	58
12	777	107
11	1117	209
10	1474	344
09	1873	509
08	1986	693
07	2084	960

Table 3: Centroids identifying misspellings in experiment 2

word length through 18	misspellings	occurrences of misspellings	incorrect replacements	occurrences of incorrect replacements
18	0	0	0	0
17	1	1	0	0
16	6	7	0	0
15	9	10	0	0
14	56	68	5	6
13	174	213	9	10
12	409	548	20	34
11	905	1267	54	75
10	1694	2282	111	149
09	2928	4379	245	508
08	4767	7724	458	1078
07	7311	13571	828	2392

Table 4: Incorrect replacements of misspellings in experiment 2

can be done either by choosing terms from documents to enrich the dictionary or by loading previously corrected text initially into the text retrieval system and then augmenting the dictionary with these indexed terms. We augmented our dictionary with a list of over 96,000⁶ LSS words that were compiled during the processing of the original LSS prototype database[9]. In addition, the list of low-count words was spell-checked against this augmented dictionary to remove correctly spelled words with low frequency from the set of (presumed) misspellings. The two sets, *centroids* and *misspellings*, were reconstructed and the misspellings were reclustered. Tables 3 and 4 show the results of these changes.

The augmented dictionary identified more centroids as can be seen by comparing Table 1 with Table 3. Consequently, 2,000 more misspelled terms were clustered. For comparison

⁶This is the number of words we obtained from the LSS dictionary and thesaurus which were not in ispell’s dictionary. It is somewhat misleading since many words in our auxiliary LSS dictionary were simply alternative forms of words that were in ispell’s dictionary. We do not know the actual number of new (unique) words gleaned from the LSS sources but the figure is significantly lower than 96,000.

misspellings	clustered with	local info selection	confusion matrix selection
ariation	aviation variation	variation	
downwar	downwarp downward	downward	
ountain	fountain mountain	mountain	
llocation	allocation location	location	
constructiona	constructional construction	construction	
transporation	transpiration transportation	transpiration transportation	transportation

Table 5: Results of local info and confusion matrix centroid selection

purposes, only the original list of misspellings from experiment 1 (correctly spelled words removed) were reclustered in the second experiment.

Error correction via approximation and dictionary lookup generally leads to ambiguity. In other words, we may have more than one candidate (centroid) for a particular misspelling. For example, the misspelling **ariation** was clustered with both **aviation** and **variation**. In the next section, we will explain two strategies which help identify the correct centroid for a multiply clustered misspelled term.

4 Localized Information and Confusion Matrix

Misspellings that were correctly clustered in experiments 1 and 2, but clustered with more than one centroid, were counted as penalties (errors). We considered these cases penalties because the process was unable to equate a misspelling to a single centroid. The first two columns of Table 5 show a set of multiply clustered misspellings and their corresponding centroids.

One of the strategies to identify the correct cluster for such misspellings is to first locate the document in which the misspelling occurs. Then, for that document, the frequency counts of the corresponding centroids are used to identify the correct centroid. For example, for the misspelling **downwar**, its centroids are **downwarp** and **downward** with frequency counts of 0 and 18, respectively. All centroids with a frequency count of 0 are eliminated. If this reduces the number of centroids to exactly 1, then that centroid is selected. In the case of the misspelling **downwar**, the correct centroid is **downward**. This strategy is an example of how our post-processing system makes good use of an inverted file system. The frequency of words and their locations in the collection play a key role in this procedure.

This first strategy, which we call *local info*, cannot always identify the correct centroid. For example, the correct centroid for the misspelling **transporation** cannot be determined (Table 5). This correction cannot be made because in the document in which the misspelling occurs, more than one of the misspelling’s centroids occurs in the document. Again, we are faced with deciding which of several centroids is the correct one for a given misspelling. The next strategy will consult the *confusion matrix* to decide on the correct centroid.

Errors	Correct	Generated
137	i	l
109	i	l
48	e	c
41	t	[
28	r	t
25	c	e
24	e	a
21	i	t
18	m	rn
17	l	i
16	2	Z
16	t	r
6	l	ϵ
3	ϵ	n
2	t	ϵ

Table 6: Confusion matrix sample

The confusion matrix is simply a table with information pertaining to errors likely to be made during the OCR process. Table 6 shows a portion of the information in the matrix. The matrix shows correct strings, OCR generated strings, and the number of times the OCR device made this error. Presently, we are only interested in operations of edit distance one—the inserting and deleting of a single character or the substitution of one character for another. The matrix represents the null string with an epsilon (ϵ).

The matrix allows for the arbitrary deletion of a **t**; however, it does not allow for the substitution of an **o** for an **i**.⁷ Using this information, we can find the correct centroid for the misspelling **transportation**. Clustering with the centroid **transpiration** would require the substitution of an **o** for an **i**, which is not allowed. The centroid **transpiration** is therefore eliminated from consideration. But the arbitrary deleting of a **t** is allowed, so the centroid **transportation** is retained. Since all other centroids have been eliminated, **transportation** is chosen as the centroid for the misspelling **transportation** (Table 5).

Table 7 shows the results from experiment 3. Again, this table lists the number of misspellings at each word length. It also shows the erroneously chosen centroids for these misspellings and their number of occurrences. Table 8 compares the percentage of corrections, based on number of occurrences, made in all three experiments. A marked improvement can be seen in each experiment phase.

Our preliminary result showed that the approximate matching of longer words is a good learning tool. Our system collects information and builds the confusion matrix while the longer words are being processed. It should be noted that the confusion matrix built reflects errors made by the device used. Consistent device errors will inherently be a part of the confusion matrix.

As mentioned, for each unique error listed in the confusion matrix, the matrix also contains information about the number of times the OCR device made the error. Using this information as the probability of each substitution, insertion, or deletion occurring, the confusion matrix selection phase will take a misspelling m along with centroids c_1, c_2, \dots, c_n

⁷The entire matrix is not shown due to its size.

word length through 18	misspellings	occurrences of misspellings	incorrect replacements	occurrences of incorrect replacements
18	0	0	0	0
17	1	1	0	0
16	6	7	0	0
15	9	10	0	0
14	56	68	1	1
13	174	213	3	3
12	409	548	10	22
11	905	1267	31	48
10	1694	2282	65	93
09	2928	4379	148	288
08	4767	7724	276	632
07	7311	13571	575	1827

Table 7: Incorrect replacements of misspellings in experiment 3

word length through 18	experiment 1	experiment 2	experiment 3
18	0	0	0
17	50	100	100
16	87	100	100
15	90	100	100
14	75	91	99
13	87	95	99
12	87	93	96
11	89	94	96
10	87	93	96
09	85	88	93
08	83	86	92
07	77	83	87

Table 8: Percentages of correctly replaced misspellings

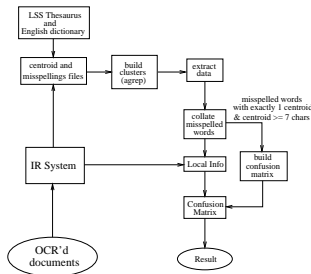


Figure 1: The global correction process

and decide which c_i 's are closest to m based on some threshold probability t . If there is exactly one centroid whose probability is greater than t , then that centroid is selected to replace m . Otherwise, the confusion matrix phase passes the misspelled word and all of its centroids to the next selection phase. If more than one centroid has a probability above t , it would be prudent to delete all other centroids with probabilities below t ; however, we have decided to take the conservative approach at this time and pass all centroids on to the next level of process selection.

Figure 1 shows roughly how the overall global correction process works. OCR'd documents are loaded into a retrieval system, and using the LSS thesaurus[9] and a dictionary[2], the lists of centroids and misspelled words are created. Next, we perform approximate pattern matching on each centroid using *agrep*[15] which creates the clusters. The information is collected, collated, and then given to the local info process and subsequently to the confusion matrix.

5 Conclusion and Future Work

So far, we have only processed words of length seven and up. Even though most of the searched words in this database are of length greater than six, we have intentions of processing words of length less than seven. With shorter words, there are many acronyms and proper names that are not found in the dictionary and therefore will inevitably be incorrectly clustered. We are currently developing procedures that will handle these kinds of special terms.

We are currently considering a misspelling and a term to be similar when they are edit distance one away from each other. We have a scheme for moving beyond this restriction

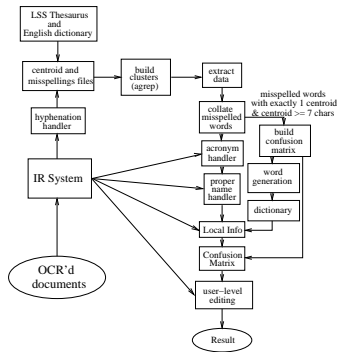


Figure 2: The process with future improvements

using the confusion matrix. Instead of using the matrix as a simple lookup table, we intend to reverse the operation and use the matrix to *generate* words. We will take a misspelling and consider all the operations listed in the matrix which apply and generate words according to these operations. For example, the misspelling *IVevada* will generate a list of terms, including the term *Nevada*, since the confusion matrix allows the substitution N for IV. This is one process which can help identify the misspellings of edit distance greater than one. These generated words will be run through the augmented dictionary to make a list of possible centroids. We can use the local info process to isolate the correct centroid if more than one is generated.

Finally, if after all processing has been performed and the correct centroid for a misspelling has not yet been isolated, the misspelling and its centroids are passed to a user-level editing interface. Figure 2 shows the flow of these planned processes.

6 Acknowledgments

We would like to thank Dr. George Nagy and Dr. Bruce Croft for their careful reading of our paper. Their suggestions have made a major improvement to our original draft.

References

- [1] Richard C. Angell, George E. Freund, and Peter Willett. Automatic spelling correction using a trigram similarity measure. *Inf. Proc. and Management*, 19(4):255–261, 1983.
- [2] R. E. Gorin, Pace Willisson, Walt Buehring, Geoff Kuenning, et al. Ispell, a free software package for spell checking files. The UNIX community, 1971. version 2.0.02.
- [3] Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *ACM Computing Surveys*, 12(4):382–402, December 1980.
- [4] T. Ito and M. Kubota. A retrieval system for on-line English-Japanese dictionary. In *Proc. 10th Intl. ACM/SIGIR Symp. on Research and Development in Information Retrieval*, pages 181–186, New Orleans, LA, June 1987. ACM Press.
- [5] James L. Peterson. Computer programs for detecting and correcting spelling errors. *Communications of the ACM*, 23(12):676–687, December 1980.
- [6] Joseph J. Pollock and Antonio Zamora. Automatic spelling correction in scientific and scholarly text. *Communications of the ACM*, 27(4):358–368, April 1984.
- [7] Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. A report on the accuracy of OCR devices. Technical Report 92-02, Information Science Research Institute, University of Nevada, Las Vegas, March 1992.
- [8] Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. An evaluation of OCR accuracy. Technical Report 93-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1993.
- [9] Science Applications Intl. Corp. Capture station simulation: Lessons learned, Final Report, for the Licensing Support System, November 1990.
- [10] Dennis Shasha and Tsong-Li Wang. New techniques for best-match retrieval. *ACM Transactions on Information Systems*, 8(2):140–158, April 1990.
- [11] R. Sinha and B. Prasada. Visual text recognition through contextual processing. *Pattern Recognition*, 21.5:463–479, 1988.
- [12] Kazem Taghva, Julie Borsack, Allen Condit, and Srinivas Erva. The effects of noisy data on text retrieval. *J. American Soc. for Inf. Sci.*, 45(1):50–58, January 1994.
- [13] H. Takahashi, N. Itoh, T. Amano, and A. Yamashita. A spelling correction method and its application to an ocr system. *Pattern Recognition*, 23(3/4):363–377, 1990.
- [14] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, January 1974.
- [15] Sun Wu and Udi Manber. Fast text searching allowing errors. *Communications of the ACM*, 35(10):83–91, October 1992.