5 Fuzzy matching and spelling correction

5.1 Introduction

Information science has traditionally been concerned with methods of storing and accessing words so that classes of orthographically similar words can be retrieved.

As long ago as 1961 Bourne and Ford published a review called "A study of methods for systematically abbreviating English words and names" [1]. Since then many other techniques have been described and tested. These techniques were reviewed by Hall and Dowling in 1980 [2]. We describe these techniques as *word representation devices* as they all share a similar function: to represent a word in such a way as to facilitate the retrieval of orthographically similar words. These devices can be used for the retrieval of misspelt words, but have also been successfully used to broaden retrieval.

5.2 N-grams

5.2.1 Definition and applications

An *n-gram* is a substring of a word, where **n** is the number of characters in the substring. Digrams, trigrams and tetragrams have been used. The assumption is that words which have a high proportion of n-grams in common will be similar. Raising the threshold for the proportion of n-grams in common increases precision but decreases recall, and *vice versa*. The length of the n-gram strings which are used will also influence recall and precision; the longer the string, the smaller the number of words which will contain that string. At the other extreme is the "1-gram" ("monogram"?).

N-gram representation has two applications. First, it can be used for spelling error detection. The word "sociology" produces this set of tri-grams: "-so", "soc", "oci", "cio", "iol", "olo", "log", "ogy", and "gy-". Its misspelling "socialogy" produces "-so", "soc", "oci", "cia", "ial", "alo", "log", "ogy", and "gy-". Since six out of nine trigrams are identical, one could assume that the two words are related as the orthographical similarity is high.

Second, it can be used to detect words which are morphologically similar in the hope that they will also be semantically related. A trigram representation of the word 'cognition', for example, produces the trigrams: '-co', 'cog', 'ogn', 'gni', 'nit', 'iti', 'tio', 'ion', 'on-'. 'Cognitive' produces '-co', 'cog', 'ogn', 'gni', 'nit', 'iti', 'tiv', 'ive', and 've-'. Again, since six out of nine trigrams are common to the two words, one could assume that the two words are related.

There are, however, certain disadvantages to the use of n-grams. Words which are orthographically very similar can be semantically dissimilar. Freund and Willett [3] cite the example of 'running' and 'cunning' (180). .In other instances, words may be retrieved, which, through the addition of affixes, are opposite to the query term (for example 'appear' and 'disappear').

A useful survey of previous n-gram experiments is provided by Zamora, Pollock and Zamora [4]. The first instance of n-gram analysis proper was published by Adamson and Boreham [5]; they based their work on the assumption that 'the character structure of a work is so related to its semantic content as to make this a useful basis for automatic classification of words'. Inter-word similarity coefficients were used as a basis for a clustering of words; they found that this led to intuitively reasonable groups of words. A small group of mathematical titles were clustered by their constituent digrams. (This procedure was tested by Willett [6]; he found that it gave poor results with the Cranfield test collection.)

5.2.2 Use of n-gram techniques to improve retrieval

Lennon and others [7] created a dictionary from the terms in the titles of the Cranfield test collection; these terms were then represented by lists of constituent di-grams or trigrams. Each of the terms in the 225 experimental queries were then similarly represented by di-grams and trigrams and matched against each of the words in the index. Index terms with a similarity coefficient greater than some threshold value were considered to be variants of the query term and automatically added to the query; the expanded queries were then used for searching the file of documents in the normal way. Retrieval experiments gave a level of retrieval effectiveness which was at least comparable with the levels obtained from a range of conventional stemming algorithms.

Freund and Willett [3] adapted the inverted file structure described by Noreault and others [8] which demonstrated that such a structure could be used for the calculation of similarity co-efficients and for the production of ranked output. Freund and Willett used a dictionary of 12,000 terms; each entry in the inverted file consisted of an n-

gram and a pointer to a list which contained the term numbers for each occurrence of the n-gram in the inverted file. The procedure separates the search word into its n-grams, identifies the appropriate lists in the inverted file and, finally, "adds" or merges the lists in order to identify the number of n-grams common to the query term and to each of the words in the file.

Freund and Willett used the Dice similarity coefficient to compare the similarity of the search terms with the index terms. These index terms were then presented to the user for possible use as query expanders. In their tests, even with the lowest threshold, there were usually less than 20 words for the user to choose from. Freund and Willett accept that this would be unwieldy with a larger file such as a library catalogue [3, p183].

5.2.3 Use of n-gram techniques to detect spelling errors

Several different methods have been used.

N-GRAM FREQUENCY TABLES

Morris and Cherry [9, 10] extracted digrams and trigrams from text words and used them to create frequency tables. The text words were then checked against a small dictionary of common words collected from about one million words of technical text. The statistics were used to calculate an index of "peculiarity" for the unmatched words and used to rank the unmatched words on the basis that those most likely to be misspelt would appear at the start of the list.

Cornew [11] also used digram frequency tables to convert an unknown text word to the dictionary word it most closely matches. The new word is then looked up in the dictionary and the process repeated until a valid word is found.

A similar method has been used by Ullmann [12]. He used n-grams to convert each unknown word to the most similar dictionary word; this method can find all dictionary words that differ from a given word by up to two errors. An n-gram method is also given for correcting up to two substitution, insertion, omission and transposition errors without doing a separate computation for every possible pair of errors. Its application is limited, however, as it is described only for six-letter words and it is dependent on the use of special hardware.

PERMISSIBLE SYLLABLE SEQUENCE COMPARISON

Nussbaum and Schek [13] used automatically generated tables for error detection which describe permissible syllables and syllable sequences based on clusters of acceptable initial and terminal letters.

INVALID TRIGRAM DETECTION

Much useful work has been done as part of the SPEEDCOP project on spelling error detection and correction [4, 14, 15, 16].

The method of trigram spelling correction used by the SPEEDCOP team differed from previous work in that it used direct measures of the trigram error probabilities rather than relying on the frequency of of trigrams.

The rationale behind the SPEEDCOP experiment was that miskeyings and misspellings would contain invalid trigrams. A study [4] was designed to determine if there is sufficient difference between the trigram compositions of correct and miskeyed words for keying errors to be detected. The motivation was the fact that if word boundaries are included there are 18,954 possible trigrams using the English alphabet, but only a small proportion of these trigrams actually occur in text. Hence it is a reasonable assumption that many misspellings will contain invalid trigrams. In fact many misspellings do not contain "invalid" trigrams - the transposed miskeying "dictoinary" for example. Experiments revealed, moreover, that the method gave inadequate recall and precision whatever the threshold chosen.

The authors suggest that it might be better to use syllabic n-grams rather than trigrams. Using positional and co-occurrence information about trigrams could also improve precision and recall. This would have the advantage that it may be possible to determine the position of an error in a misspelling. A word is assumed to be misspelt if it contains two consecutive trigrams with error probabilities greater than some threshold. This method determines the error location accurately to within one character in 94% of instances, although it cannot distinguish accurately between different error types.

POSITIONAL N-GRAM ANALYSIS

Riseman and Hanson examined the effectiveness of various methods of using contextual information to detect and correct keyboard errors [17]. They used positional "binary" n-grams to detect miskeyed words, establish the position of the error and in some cases to determine the character which can correct the word.

Riseman and Hanson contend that while positional n-grams require more storage, the process of collecting and storing contextual information is simplified and computational complexity is reduced. Moreover, positional information is more compact than probability information about n-grams.

Carlson [18] used positional trigram probabilities to correct errors in English first names and fix the position of the error; an error correction rate of 95% was achieved.

5.2.4 *Different values of n*

The value of **n** has a strong influence on all n-gram techniques.

If n-grams are to be used in a spelling error *detection* system then a high value of **n** is more likely to make erroneous spelling produce "peculiar" n-grams. This is certainly not always the case ("sociology" generates acceptable n-grams for any value of **n**). But if the object is to find as many words as possible of which this may be a misspelling, then **n** should be one or two.

The problem is different if n-grams are being used to find words which are morphologically similar. In this case, the value of **n** should presumably be rather close to the average length of syllables: i.e. **n** should be two, three or four.

In practice, computing storage and processing requirements are an important factor. Substantial storage is needed if an additional index of "long" n-grams is to made from an initial inverted index. Several experimenters have used trigrams because they represent a compromise between di-grams which are often inadequately "strong" and tetragrams, of which there are a great many.

5.2.5 *Effectiveness tests*

Lennon and others [7] evaluated the effectiveness of a similar technique to that used by Adamson and Boreham [5]. Index terms with a similarity coefficient greater than a threshold value were considered to be variants of the query term and automatically added to the query; the expanded queries were then used for searching the file of documents in the normal way. Retrieval experiments demonstrated that this procedure gave a retrieval effectiveness which was at least comparable with that obtained with a range of conventional stemming algorithms.

N-gram measures seem usually to have been used in experiments with rather small files. The method used by Lennon and others was reliant on the matching of the search term with every term in the file.

The technique used by Freund and Willett [3] performed with reasonable accuracy on their 12,000-word dictionary but it would probably need to be modified if it were to be used in a much larger library catalogue. Freund and Willett point out that the use of trigrams can lead to an unacceptably large number of non-related items, especially if a low similarity threshold is used. They feel however that "the

numbers of indexed terms retrieved using trigrams is quite acceptable for rapid visual inspection at a terminal" [3, p182]. Using digrams rather than trigrams sometimes retrieved a very large number of words.

Experiments conducted with the SPEEDCOP system demonstrated that the invalid trigram method gave inadequate recall and precision whatever threshold is chosen. The authors suggest that the adoption of a more sophisticated error detection measure which might use syllabic n-grams rather than trigrams or use positional and co-occurrence information about trigrams could improve precision and recall. Trigram analysis has the advantage over the use of a dictionary in that it is sometimes possible to determine the position of an error in a word. This is inherent in the method used since a word is defined as misspelt if it contains two consecutive trigrams with error probabilities greater than the threshold selected. The trigram analysis method determines the error location accurately to within one character in 94% of instances, although it cannot distinguish accurately between different error types.

Riseman and Hanson [17] compared a binary positional trigram correction procedure with a dictionary lookup procedure. They used a fairly large set of six letter words. The positional trigram method was not quite so effective in correcting errors as the use of a complete dictionary, but the detection and correction rates were high enough that the difference was marginal. They conclude that, if the dictionary is fairly large, the trigram method is computationally faster and occupies less storage. However, they assume that the entire dictionary has to be searched. This is only the case if no assumptions are made about the type and nature of the errors. In practice, dictionaries are stored in such a way that comparatively small lists of candidate corrections for most erroneous words can be found rather quickly.

5.3 Soundex, soundex-type and other abbreviation codes

5.3.1 *Definition and applications*

Soundex was patented as a clerical technique for the manual coding of names. It was designed to help in the retrieval of misheard or misspelt names. There have been many modifications of the original Soundex procedure for different applications, and several programs have been published [19, 20, 21]. The name has come to be used for a wide variety of word representation techniques. (When used generically we write it as "soundex" by analogy with "hoover"). Unlike n-grams, which represent a word by a set of character strings (and so greatly *expand* the original number of characters), soundex-type codes represent a word by its most significant characters (and so *reduce* the original number of characters). The original Soundex

represented names phonetically. It retained the initial letter, removed vowels and a few other letters, replaced consonants by codes for phonetically related groups, removed repeated codes, and finally truncated the name at four characters. Many, but not all, soundex-type procedures are also phonetically based.

The most appropriate application for a soundex type representation code is in finding candidate replacements for misspelt and miskeyed words.

5.3.2 *Use of Soundex-type codes: a survey*

Tests conducted by the Dominion Bureau of Statistics of Canada [22, 23] demonstrated that while Soundex compared favourably to other codes it did not perform adequately with non-Western names. Soundex was designed to retrieve words after errors in hearing rather than keying and spelling errors. Even so, it is easy to find names which present retrieval problems - "Rogers" and "Rodgers" for example. Fenichel and Barnett [24], quoting Alberga [25], point out that there is some evidence that written spelling errors are often misconstructed from the correct forms by the same errors as phonetic errors of hearing.

Davidson [26] used a soundex-type algorithm to encode the names of airline passengers in order to cope with misheard names. The Davidson code is not phonetic: it was felt that the international scope of the names to be included would make the phonetic equivalences of certain letters difficult to standardise. Apart from this it is almost identical to Soundex, except that there is a fifth character which contains the initial of the first forename, if known.

Blair [27] tried a soundex-type coding scheme which aimed to retain the differing amounts of information associated with different relative positions in the word and with different letter frequencies. The highest weight is given to the first letter, followed by the last letter, the second letter, the next to the last letter and so on. Each letter was scored by combining its positional score with its letter-frequency score. The "least important" letters were then deleted until the required code length was reached.

Blair's code correctly identified 89 out of 117 misspelt words and incorrectly identified two. Errors arose either because the word was not in the original vocabulary or because the misspelling was so extreme that it gave rise to a different abbreviation. Blair suggests correcting the first type of error by adding it to the vocabulary when it is updated. The second type of error is corrected by creating a special index in which the correct spelling of a problem word, and its abbreviation, are identified by a link.

Damerau's technique for the computer detection and correction of spelling errors is described in [28]. His method assumes that a word which cannot be found in a dictionary has at most one error; inspection of keyboard errors in a retrieval system showed that over 80% of errors were caused by a wrong, missing or extra letter or a single transposition. His procedure works by assuming that any one of these errors might have occurred. It reverses all possible errors of these types in unidentified words until a dictionary match is or is not found.

Tests with a collection of common spelling errors (of the four types above) gave a success rate of over 95%. Damerau compared his technique to that of Blair and, for the corpus of common misspellings, similar results were obtained. Blair's technique, perhaps not surprisingly, could not deal with machine-garbled text. The computational cost is not given but is likely to be substantial given the extensive letter comparisons which have to be made.

Bourne and Ford [1] summarised different methods for abbreviating words systematically. They tested the performance of thirteen basic techniques in the retrieval of technical documents from a collection at the Stanford Research Institute. This is still a useful survey of abbreviation techniques, but they did not consider the effectiveness of the techniques either for matching misspellings or for increasing recall.

For matching misspelt personal names, Greenfield [29] compared the Soundex code, the Davidson code, exact name searching and a search on secondary characteristics such as date of birth, sex or age. The test employed was that of finding duplicate records in a database.

Of the codes, Davidson gave a slightly better hit to mismatch ratio than even the exact surname technique did, and gave 2.3 times fewer mismatches than did Soundex. Soundex did produce a slightly higher number of true matches than Davidson but the difference was negligible. Greenfield concludes that '...Davidson's is the technique of choice' [29, p233], but he makes suggestions for improving its performance. For example, Davidson missed fourteen matches correctly made by Soundex: of these, Davidson missed eight because the letters 'm' and 'n' were not merged. These results do not confirm reservations expressed by Moore [30] that Davidson would produce more false negatives than Soundex. Greenfield does not test a modified Davidson code.

5.4 Fuzzy matching in online catalogues

5.4.1 Spelling correction

Reasons for providing a method to deal with miskeyings and misspellings were given in 2.4.3. This section discusses the methods which can be used and how they should be offered.

One problem in online catalogues is when to assume that a word or phrase is misspelt. If the lookup procedure is able to determine that there is a unique key which is a near match with the user's key then there may be no need to perform any correction. This would apply particularly to phrase-matching systems. Where there is a single subject heading or name or title which matches on all except the last few characters of the user's key, this should be offered as a match (with perhaps an unobtrusive message to the effect that "this doesn't exactly match your search"). We do not know of any catalogues which can do this. It can be rather demanding on computing resources, involving stripping off final characters and shuffling around in the index.

Even with keyword access, if a title or subject word is not found but the proportion of words which have index matches is high enough, then the result of a successful AND on the words which *are* found has a reasonable chance of being the sought item. The original Okapi system would do this. Better precision may be obtained by using an inverse word frequency weighting (giving a higher weight to "rare" words). In the SWALCAP LIBERTAS system, a "notional" weight is assigned to words which are not found in the appropriate index, and a search missing a word can still succeed, although the user will be warned that the item(s) found do not match the search exactly.

Spelling correction in online catalogues should also take into account the fact that spelling errors vary according to the type of search. Transaction log analysis of Okapi '84 [31] showed a marked difference between errors in specific item searches and subject searches. In specific item searches users are often copying from a printed source. In particular, there are rather few errors in personal names, and they are more likely to be phonetic or *spelling* mistakes than *keying* mistakes. In subject searching miskeyings predominate, and they are frequent.

The simplest method of dealing with search terms which are not found is to report a failed search, leaving it to the user to re-enter the search appropriately. Many users do not notice that they have made a mistake and often leave the catalogue assuming that the sought item or subject is not in it. Hence this option, or "non-option", which is what most current keyword-type catalogues provide is

insupportable.

The next level is to provide a specific message about each word which was not found. Users still have to re-enter their search, but at least they know why it failed. A few current catalogues do this.

If a suitable fuzzy matching procedure is available the same result can be reported to the user together with an option of looking for "similar" words (or names). This was tried, although not on "real users, in an intermediate version of Okapi.

5.4.2 Spelling correction using n-grams

The work of Freund and Willett [3] has been discussed in 5.2.5 and 5.2.6 above. It was primarily designed to improve recall by using n-grams to retrieve variations of root forms. N-gram representation has been used, however, in the detection and correction of spelling errors, notably in the SPEEDCOP project. This project has considered both the use of n-gram analysis and dictionary look-up for spelling correction purposes. The use of dictionaries will be considered later as this is technically different from the use of algorithms.

None of the experiments which have been described have been applied to library catalogues. The SPEEDCOP experiments were used in the batch editing of chemical information. There are at least two important ways in which spelling correction in an online catalogue differs from the SPEEDCOP environment. First, correction has to take place in real time with limited computing resources. N-gram techniques which need a large amount of computer storage space and processing time are unlikely to be adaptable for use in an online system. Secondly, it is possible that this technique is particularly well suited to a "hard language" scientific discipline where the language is generally well structured and unambiguous; n-gram analysis of the language of other fields of knowledge may might be considerably less profitable.

It should be noted that even with the advantages of working with hard language information, and being able to disregard the considerable problems of designing a suitable interaction, the method gave inadequate recall and precision whatever the value of the matching threshold. The authors' suggest that the adoption of syllabic n-grams rather than trigrams, or the use of positional and co-occurrence information about trigrams, would improve precision and recall. However, the latter would probably increase storage requirements.

To summarise, although n-grams have been used experimentally, it is not at all certain that they are suitable

for use in the online catalogue of a general library.

5.4.3 Soundex-type codes in online catalogues

Unlike n-gram techniques, soundex-type devices have been used in experimental online catalogues.

The Bibliographic Access and Control System (BACS) developed at the Washington University School of Medicine Library includes a facility which will look for approximate spellings in the author, title, subject and series fields. If no records are found from an implied Boolean AND, then the system automatically looks for approximate spellings. It uses a simple soundex-type algorithm which drops trailing "s", doubled consonants and vowels other than initial vowels [32].

This procedure is automatic and displays a helpful message to the user "Trying approximation search" before showing records which contain approximately matching words. BACS is notable for its simplicity and its cordiality, making few demands on the user. No tests on its effectiveness appear to have been conducted. BACS is used in a medical library. Medical language may be particularly appropriate for soundex-type processing as it is regular in word construction and unambiguous in application. Problems of spelling correction will be more complicated in a general academic library serving a much wider population and covering a wider range of subjects areas.

The retrieval system which has been developed at Massachusetts General Hospital [24] also attempts to correct spelling and typing errors. If no match is made even after the user's term has been stemmed then a two phase process is used. This process first identifies similarly spelt words in the database, and then interacts with the user in order to see if any of these words were the intended word. It uses a soundex-type scheme which deletes vowels, "singles" repeated consonants and conflates similar sounding consonants to a canonical representative of that class. This is supplemented by a dictionary of common misspellings, alternative spellings, and non-preferred terms (which are stored but not displayed to the users to avoid encouraging their use). The dictionary also includes obscenities which are stored as terms so that they can be ignored and not printed on the screen.

Even after approximate matching, the process does not demand an exact match; a match is achieved if the first few characters are identical and the length matches to within a margin of 20%. The user may specify that the search be restricted to certain indexes (it can distinguish, for example, between the names of drugs, anatomical terms, laboratory test, and therapies). The process finds approximate matches for about half of the searches which fail to

find an exact or stem match.

The authors report that in most instances only one term is found but occasionally up to twelve matches are made. When this happens, the user is asked which term (if any) is correct. About 60% of the suggested searches are accepted by the users although the logging procedure cannot estimate what proportion of these successes correspond to the users' original intent and what proportion are accepted as alternative and novel terms. A review of use over four months demonstrated that 92% of searches found matches of some sort at one of the three stages.

The authors do not state whether library staff or users regard a success rate of about half as being satisfactory, nor a percentage figure for searches which locate more than one match. Both of these factors will be important for the overall success of the facility.

The Paperchase system at Beth Israel Hospital in Boston also tries to be tolerant of spelling errors. One of the Paperchase design objectives was to allow the user to enter whatever seems natural; this was based on the preponderance of abbreviations and acronyms in medical literature. The system attempts to match what is typed to items in the database by applying methods which a person might use. The main technique is to look for partial matches on words; thus NEW EN JOUR MED would retrieve NEW ENGLAND JOURNAL OF MEDICINE after attempting this matching technique. If the truncation of words fails then the order of words is made fuzzier and individual words in assumed phrases are rotated [33].

A different and more sophisticated approach is taken by the LEXICON information retrieval system [34]. It uses a two-step procedure consisting of a modified Soundex system (with a relatively low threshold) followed by a high threshold system. This should automatically correct 60-70% of the errors encountered [35]. This correction rate is slightly better than that achieved at Massachusetts General Hospital ("about half").

An alternative application of Soundex has been suggested for use in a library which is moving away from the concept of uniform headings. It has been compared to standard methods of authority control: "Sound based coding focuses on the similarities found among the many forms of a name that a person might use when searching a database and brings them together for a searcher's perusal" [36, p132]. This sounds rather like an attempt to avoid proper cross referencing. Before assuming that a user's word or name is a mistake, it should be looked up in a list of "non-preferred" forms. If found, the search should auto-matically be directed to the records indexed under the "see" reference.

References

1 BOURNE C P and FORD D F. A study of methods for systematically abbreviating English words and names. *Journal of the Association for Computing Machinery 8*, 1961, 538-552.

2 HALL P A V and DOWLING G R. Approximate string matching. *Computing Surveys 12* (4), December 1980, 381-402.

3 FREUND G E and WILLETT P. Online identification of word variants and arbitrary truncation searching using a string similarity measure. *Information Technology : Research and Development 1*, 1982, 177-187.

4 ZAMORA E M, POLLOCK J J and ZAMORA A. The use of trigram analysis for spelling error detection. *Information Processing and Management 17* (6), 1981, 305-316.

5 ADAMSON G W and BOREHAM J. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval 10*, 1974, 253-260.

6 WILLETT P. Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms. *Journal of Documentation* 35 (4), Dec 1979, 296-305.

7 LENNON M *and others*. An evaluation of some conflation algorithms for Information Retrieval. *Journal of Information Science* 3, 1981, 177-183.

8 NOREAULT T, KOLL M and McGILL M J. Automatic ranked output from Boolean searches in SIRE. *Journal of the American Society for Information Science 28*, 1977, 333-339.

9 MORRIS R and CHERRY L L. Computer detection of typographical errors. *Bell Laboratories Computing Science Technical Report, 18*, 1974.

10 MORRIS R and CHERRY L L. Computer detection of typographical errors. *IEEE Transactions Professional Communication PC-18* (1), 54-63.

11 CORNEW R W. A statistical method of spelling correction. *Information Control 12*, 1968, 79-93.

12 ULLMANN J R. A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. *Computer Journal 20*, (2), 1977, 141-7.

13 NUSSBAUM R and SCHEK H J. *Automatic error detection in natural language words* (Report TR 78.06.005). IBM Heidelberg Scientific Center, 1978.

14 POLLOCK J J and ZAMORA A. Collection and characterization of spelling errors in scientific and scholarly text. *Journal of the American Society for Information Science* 34 (1), Jan 1983, 51-58.

15 POLLOCK J J and ZAMORA A. System design for detection and correction of spelling errors in scientific and scholarly text. *Journal of the American Society for Information Science* 35 (2), 1984, 104-109.

16 ZAMORA A. Automatic detection and correction of spelling errors in a large data base. *Journal of the American Society for Information Science* 31 (1), 1980, 51-57.

17 RISEMAN E M and HANSON A R. A contextual postprocessing system for error correction using binary n-grams. *IEEE Transactions on Computers* C-23 (5), May 1974, 480-493.

18 CARLSON G. Techniques for replacing characters that are garbled on input. In: *1966 Spring Joint Comput. Conference, AFIPS Conference Proceedings* 28, 1966. Washington DC : Spartan, 1966, 189-192.

19 MUNNECKE T. Give your computer an ear for names. *BYTE* 5, May 1980, 196-200.

20 CLARKE D. Sounds familiar. *Practical Computing* 7, February 1984, 90-93.

21 JACOBS J R. Finding words that sound alike : the Soundex algorithm. *BYTE* 7, March 1982, 473-474.

22 SUNTER A B. *A Statistical Approach to Record Linkage* Ottowa : Dominion Bureau of Statistics, 1967.

23 SUNTER A B and FELLEGI I P. *An Optimal Theory of Record Linkage*. Ottowa : Dominion Bureau of Statistics, 1967.

24 FENICHEL R R and BARNETT G O. An application-independent subsystem for free-text analysis. *Computers and Biomedical Research* 9, 1976, 159-167.

25 ALBERGA C N. String similarity and misspellings. *Communications of the ACM* 10 (5), May 1967, 302-313.

26 DAVIDSON L. Retrieval of misspelled names in an airlines passenger record system. *Communications of the ACM* 5 (3), March 1962, 169-171.

27 BLAIR C R. A program for correcting spelling errors. *Information and Control* 3, 1960, 60-67.

28 DAMERAU F J. A technique for computer detection and correction of spelling errors. *Communications of the ACM 7* (3), March 1964, 171-176.

29 GREENFIELD R H. An experiment to measure the performance of phonetic key compression retrieval schemes. *Methods of Information in Medicine 16* (4), 1977, 230-233.

30 MOORE F J. Mechanizing a large register of first order patient data. *Methods of Information in Medicine 4*, 1965, 1-10.

31 JONES R M. Improving Okapi : transaction log analysis of failed searches in an online catalogue. *VINE 62*, May 1986, 3-13.

32 KELLY B *and others*. Bibliographic Access & Control System. *Information Technology and Libraries 1* (2), June 1982, 125-132.

33 COCHRANE P A. "Friendly" catalog forgives user errors : no librarian intervention necessary on dream online system called "Paperchase". *American Libraries 13* (5), May 1982, 303-306.

34 JOSEPH D M and WONG R L. Correction of misspellings and typographical errors in a Free-Text Medical English Information Storage and Retrieval System. *Methods of Information in Medicine 18* (4), 1979, 228-234.

35 WONG R L *and others*. Profile of a dictionary compiled from scanning over one million words of surgical pathology narrative text. *Computers and Biomedical Research 13*, 1980, 382-398.

36 ROUGHTON K G and TYCKOSON D A. Browsing with sound : sound-based codes and automated authority control. *Information Technology and Libraries* 4 (2), June 1985, 130-136.