Preprint No. 7 Classification: IR 4.1

Automatic error-correction in natural languages

by

A.J. Szanser
Computer Science Division,
National Physical Laboratory, Teddington, England.

Abstract

Automatic error-correction in natural language processing is based on the principle of 'elastic matching'. Text words are segmented into 'lines' with letters arranged according to a pre-determined sequence, and then matched line-by-line, shifts being applied if the numbers of lines are unequal.

In order to resolve the possible multiple choices produced, the method may be supplemented by another one, based on the observed repetition of words in natural texts, and also by syntactic analysis.

This paper describes the above methods and gives an account of an experiment now in progress at the National Physical Laboratory.

1. Elastic matching

With increased application of computers in the processing of natural languages comes the need for correcting errors introduced by human operators at the input stage.

A statistic investigation [1] revealed that roughly 80 per cent of all misspelled words contain only one error, belonging to one of the following cases: a letter missing, an extra letter, a wrong letter and finally two adjacent letters interchanged. As such an error can occur in any position, a check by trying all possible alternatives in turn is clearly impracticable.

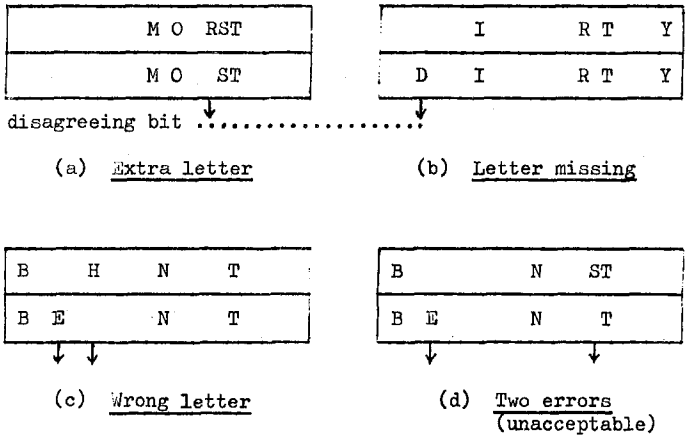
A method which can obtain the same result but in a less tedious and time-consuming way has been worked out and experimented upon at the National Physical Laboratory, Teddington, England. This method, named 'elastic matching' was first proposed at the 1968 I.F.I.P. Congress in Edinburgh, Scotland [2].

The elastic matching of words consists basically of coding all the characters (letters) as bits in a computer word, allotting to each letter a specific position. The whole English alphabet will therefore be represented by a sequence of 26 bits, although their order, as will be shown below, may, and indeed should, differ from the usual order of letters in the alphabet.

All words belonging to a complete set, which may be a list of words or a whole dictionary, are 'linearized', that is converted into segments, called 'lines', in which the letters are arranged in the agreed order. If the current letter has a position prior to the last stored, a new line must be started. Thus, if the sequence in question were the alphabet itself, the word 'interest' (for example) would be linearized as follows: 'int-er-est'. The actual sequence, by the way, has to be chosen in such a way that it would produce the longest possible lines or, in other words, the minimum number of lines for a given sample of text.

The matching is carried out not between words but between lines. All errors will then stand out immediately as one or more disagreeing bits*. In the case of two bits a simple check will reveal whether this is the result of an accepted type of error (one wrong letter, or two adjacent letters interchanged), or the result of two separate errors, and therefore to be rejected under the limit accepted (one error per word).

In the examples shown below the alphabet has been assumed to be the linearizing sequence; this is done for the sake of better clarity only.



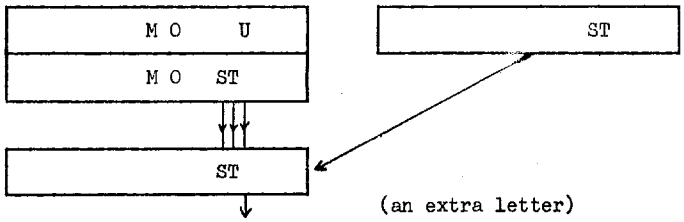
The result (d) is unacceptable because the two disagreeing bits

---

* For example by using the logical 'NOT-EQUIVALENT' operation.

are formed as a result of two errors (extra S and missing E). In the computer check this is shown by the two outstanding bits (letters) being separated by another bit (letter).

If the numbers of lines in the two versions (misspelled and correct) are unequal, the procedure is as follows. The next line of the longer version is shifted back and matched against the result (that is, the disagreeing bits) of the previous match. Thus, for example,



In the case of two disagreeing bits some simple checks have again to be made to eliminate the two-error cases, and also to prevent spurious matches resulting from the self-cancellation of characters between the two successive lines of the same version.

More particulars of the operation of this method can be found in a special paper [5].

2. The dictionary organization

The elastic matching, as was mentioned above, is applicable against any set of (correct) words, which may be, for example, a list of proper names, or any other words, even of artificial (e.g. programming) languages.

It is, however, the application to natural languages, in particular English, which is the subject of this paper. There are two problems which have to be overcome or, at least, reduced to manageable proportions before this method can be applied using a complete English dictionary.

The first problem is access to the dictionary which may contain tens or even hundreds of thousands of entries. This number, however, includes all grammatical forms of English words (fortunately, they are not so numerous as in highly inflected languages such as Russian).

The dictionary look-up takes different forms depending on the

way in which the dictionary is organized. The latter could have either a tree-like structure (preferably built of 'lines') which is likely to be quicker in operation, or a list structure, in which words may be grouped by their line numbers, then by numbers of letters and finally, if the lists are still too long, by part-alphabetization (according to the accepted sequence). This structure is easier to prepare.

The words to be checked against the dictionary of the list structure will be linearized, and during this process the numbers of their lines and letters will be determined. The sections of the dictionary to be used in the matching process will be those with equal numbers of lines (and letters) and those immediately below and above these numbers (depending on the error threshold accepted).

The other problem is connected with the number of multiple matches likely to occur, especially for short words.

Two ways of alleviating this problem are described in the next section.

3. The supplementary procedures

3.1 The general-content check

One possibility of choosing between the multiple equivalents produced by dictionary look-up is to select those which are repeated throughout the article or speech in question. For this purpose a procedure called 'general-content check' has been devised. As the text is processed, each different word satisfying certain conditions is stored. Then all multiple results from dictionary look-up are compared with the contents of this store (which may also be organized into sections) and words found there are given preference to others. The idea behind this is, of course, that words tend to be repeated by one writer or speaker.

The size of the sample processed for the general-content check must not be too small or too large. The optimum size should be determined experimentally, but one may risk the guess that perhaps 1-2 thousand (current) test words are a practical amount.

Further, there is no need to store all the different words. Ideally, these should be the so-called 'content' words, such as nouns, verbs, adjectives and adverbs, whereas the remaining, 'function' words (prepositions, conjunctions, etc.) should be left aside, as not being content-typical. The selection can easily be done in the storing process if dictionary entries are suitably marked.

Also, if one grammatical form of a word is stored, there is no

need for storing others, so that the general-content vocabulary may assume the character of a stem-word list. This again, can conveniently be arranged both in storing and in matching.

3.2 Syntactic analysis

Another possibility of making a choice between multiple equivalents is syntactic analysis. This is especially promising, because if one considers a typical lexical set of common words, one must notice that long words (which give, as a rule, better results in elastic matching) usually belong to 'content' words, whereas the 'function' words, which are specially amenable to syntactic analysis are normally short and, therefore, would either produce more multiple choices or, if of less than four letters, would escape the elastic matching altogether*. In this way the two methods are largely complementary. More of syntactic analysis in error-correction will be said below.

Neither of the two supplementary methods mentioned above is applicable where elastic matching is used for non-textual material (list of names, etc.).

4. An experiment in automatic error-correction

An experiment has been carried out at the NPL on the lines described above.

First of all, an optimum linearizing sequence had to be established for English texts. Several methods were used for this purpose, both statistical and purely linguistic, and the results were submitted to computer tests. Sequences bringing lower yield had been gradually eliminated and changes were made in those remaining, in order to determine the optimum sequence by the well-known 'hill-climbing' technique. This investigation has been fully described elsewhere [3] and it has produced the following sequence:

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

Next, through lack of a proper dictionary, the general-content check procedure was used to compile lists of words occurring in selected stretches of English (parts of three articles on physics, linguistics and socio-politics, containing about 3,000 text words in all).

---

* This limit has been accepted.

Several hundred distorted words (based on words in the same articles) were matched against these vocabularies. After all the corrections and adjustments, the need for which naturally occurred during the tests, have been made, the final results can be summarized as follows:

(i) The retrievals were both exact and complete, in the sense that no misspelled words (within the proper error limit) were left unretrieved and no wrong retrievals were produced;

(ii) The number of multiple equivalents increased rapidly as the lower limit of the number of letters (four) in a word was approached (in some cases up to five equivalents);

(iii) The number of multiple equivalents was generally insignificant for 'content' words (in most cases only one word was retrieved), whereas 'function' words often produced many equivalents, e.g.

'THER' → THEY, OTHER, THEN, THEM, THERE, THEIR

All these observations confirmed the results anticipated in previous sections.

The latest stage of the experiment is being carried out at the time of writing this paper (May, 1969). The author is now able to use the English side of the Palantype - English dictionary* of about 80,000 entries. For the sake of economy in programming and machine time, only one section of the dictionary, namely the entries starting with letter S (about 10% of the whole dictionary) is being used. The linearization and organization of this section is now in progress. This will enable the author to test a more complete dictionary look-up than before, together with general-content check and later with syntactic analysis as well.

5. Other applications

5.1 Apart from the general use for natural English texts, an application of the elastic matching technique has been proposed in the automatic transcription of machine-shorthand of the Palantype system. This system uses a special machine with a keyboard enabling the simultaneous striking of several keys, each 'stroke' corresponding to a phonetically-based group of consonants and vowels, roughly equivalent to a syllable. In normal operation all the characters of each stroke are printed together on a continuous paper band, shifting after each

---

* This will be explained below, Section 5.

stroke. The recording is later read and transcribed by a human operator. Since the latter part of the operation is naturally much slower (about four times that of the recording), a project, now in progress at the NPL, aims at securing automatic transcription, in which the character levers, in addition to the ordinary printing action, activate electric contacts. These create impulses, which are fed into a computer and result, after a series of operations, in printing out a text as near to ordinary English, as possible. One of the problems encountered in this process is caused by the flexibility of the recording convention, enabling the human operator to record phonetic combinations in more than one way. Generally, this is provided for by inserting in the automatic Palantype-English dictionary all versions of each word that can be reasonably foreseen. In practice the unforeseen sometimes happens and the word is output untranslated (but 'transliterated' phonetically), which is at the best annoying, but may even be unreadable. An analysis has shown that most of the deviations from standard versions stored in the dictionary are caused by a few convention rules, such as e.g. 'vowel elision': any unaccented vowel in a word can be omitted. Now, if the matching is done not on Palantype strokes but on their linearized versions, the elastic matching rules can easily be adjusted to include the versions produced.

Incidentally, the Palantype sequence is already partly linearized, and reads:

SCPTH + MFRNLYOEAUI . NLCMFRPT + SH

(the "+" and "." signs have special phonetic functions). For the linearization purposes all that is needed is to exclude the repeated consonants (from second "N" to the end); the number of lines will therefore exceed the number of 'strokes'.

The relevant procedures have been fully tested on sample lists of standard and non-standard versions (containing up to 300 words) and were found satisfactory. The implementation, however, for use with the full dictionary remains to be done. It is still not clear whether it would repay to linearize and store in this form the complete dictionary of eighty odd thousand entries; or whether it would be more practical to linearize while checking, stroke by stroke, which would be, of course, a much slower procedure. At the present time it does not look likely that either solution would lead to standardization being possible in 'real-time', but there remains the possibility of an 'errata' sheet being produced almost immediately after the normal output. More particulars about this application can be found in the paper [4].

5.2 Another application, now under consideration, is the retrieval of misspelled proper names from lists used in a fact-retrieval project, which is also in progress at the NPL.

6. Further plans

Once the work on the English dictionary section is completed it is hoped that the results will be extended to the full dictionary, and give the correct idea of the size of the problem and the times involved in the operation of the system. Also the question of the most reasonable dictionary organisation will find an answer.

Independently of the above, the application of syntactic analysis for the resolution of ambiguities (multiple equivalents) will be studied. A very limited syntactic check has already been theoretically worked out and proposed for the resolution of ambiguities in the automatic transcription of machine-shorthand. This is limited to the inspection of only the adjacent words; since, however, in the Palantype system speed is all important, the limitations brought by this condition may still be worth incorporating in the system.

The work described above has been carried out at the National Physical Laboratory, Teddington, England.

References

[1] F.J. Damerau, "Technique for computer detection and correction of spelling errors", Comm. A.C.M. 7, (3), 1964.

[2] A.J. Szanser, "Error-correcting methods in natural language processing", IFIP Congress 68, Edinburgh, August, 1968 (Booklet H, pp 15-19).

[3] "Error-correcting methods in natural language processing - I. Optimum letter sequence for longest strings in English", COM.3CI. T.M. 12, National Physical Laboratory, Teddington, England, May 1968.

[4] "Error-correcting methods in natural language processing - II. Standardization of variants in the Palantype automatic transcription", COM.3CI. T.M. 16, April 1969.

[5] "Error-correcting methods in natural language processing - III. 'Elastic matching' technique in the processing of English", COM.3CI. T.M. 21, April 1969.