[*Proceedings of the Institute of Electrical Engineers*, vol.103, part B: Supplement no.3 (1956)]

681.142

Paper No. 2142 M
Nov. 1956

COMPUTER OPERATIONS REQUIRED FOR MECHANICAL TRANSLATION

By A. F. PARKER-RHODES, M.A., Ph.D.

(The paper was received 25th April, 1956. *It was read at the* CONVENTION ON DIGITAL-COMPUTER TECHNIQUES, *12th April,* 1956.)

The completely mechanized process of translation from an extended text in a given source language into a given target language, to be realized as a practical and economic project, will require the following processes to be perfected. First, we must devise means whereby an ordinary printed text can be mechanically read, and the separate signs of which it is composed recorded in a suitable signal code. I take it that this step lies outside the field of this Convention, but it is important to bear in mind that it should be done. The employment of a human operator to encode the text, though necessary in the present stage of the research, is no solution. Even if it were economically feasible, to replace a skilled translator by an unskilled encoder would be a socially retrograde step. From this point on I shall be assuming that the input of the translation process consists of a sequence of signals, which I shall take to be binary numbers, each of which represents a printed sign or significant space in the original text. I shall not here be concerned with the still remoter prospect of auditory input.

The next stage of the process is to recognize in this sequence of signals those subsequences which we are to acknowledge as 'words'. This is in practice the hardest, though in theory the easiest, step. It has therefore received hitherto the bulk of the attention which has been given to mechanical translation. Basically, the problem is one of matching the successive units of the input with a permanent store or 'dictionary' of unit-sequences representing the 'words' of the source language, and of replacing each 'word' recognized by a conveniently compact signal for recognition during the subsequent stages of the programme. It is, of course, impossible to handle the input 'words' as such. In many European languages words may run to over a dozen letters very frequently, and each letter will need at least five and probably six binary digits, whereas few commercial computers could afford space for more than 20 digits per word, and every saving of space beyond that would speed up the process. The problem of matching between long sequences of digits is one of some difficulty, and has received a good deal of attention in America, and especially from Dr. Booth in this country. No doubt it is most efficiently done by machines designed especially for the purpose, but in any case it does not demand any great versatility from the operational point of view. To distinguish identity from non-identity (e.g. by subtraction and zero-recognition) at very high speed is all that is required. The sort of equipment necessary for this has been discussed elsewhere in this Convention, and I shall not pursue the matter further here.

When the input has been through this stage of dictionary-reading, it will emerge in a much shortened form, consisting no longer of signals representing letters but of rather longer signals, probably of about 12 binary digits each, each of which represents a 'word' of the source language. What particular units we recognize as 'words' is not always obvious. It is a matter for the linguist to determine, but it depends, not only on the nature of the language itself, but on the kind of process to which the units recognized are to be subsequently subjected, and also on the nature of the matching processes involved. The sequence of word signs produced by the preceding stage forms the input for the third stage, in which the actual translation takes place, whereby the input sequence is replaced by a new output sequence, the units of which are then converted by the fourth stage of the process into words of the target language. This last stage is comparatively simple, since it is always possible to arrange for the output signals produced by the third stage, i.e. the stage of actual translation, to be in one-to-one correspondence with words of the target language, so that the fourth stage is merely one of simple matching, replacing an arbitrary numerical symbol by a sequence of letter symbols similar to those forming the input of the second stage. The fifth stage is trivial, being the conversion of the letter symbols provided by the fourth stage into typed or printed signs by an ordinary teleprinter.

It is the third stage which provides the most interesting problems for the computer engineer, and on which the Cambridge Language Research Unit has been mainly working. For more detailed discussion it can be broken down into a number of steps. Three main things need to be done: first, to choose the word of the target language to be used to translate the given word of the text, with all the necessary complications such as allowing for meanings being influenced by context, for different words having the same spellings, for whole phrases being reconstructed so that no one-to-one translation is possible, and for the special cases of idioms and proper names. A second requirement is to recognize the significant word groups and clauses of which the text is made up, and to correlate these with their counterparts in the target language. This step has often been neglected in mechanical translation work, on the grounds that a more or less amorphous pidgin-translation would serve for practical purposes; but in fact we must always bear in mind the ultimate commercial possibilities of any process we may devise, and it is clear that for a long while mechanical translation, if feasible economically at all, will remain only marginally so, and will certainly be unable to compete with human translation if it does not produce a readable text in a single

Dr. Parker-Rhodes is with the Cambridge Language Research Unit.

process. The neglect of this stage might be allowable if it were a costly or difficult one to carry out, but in fact, as this Convention has abundantly shown, there is nothing in it beyond the technical competence of existing computer equipment or, as I shall try to explain, beyond the reach of mathematical linguistics. A third necessary step is the rearranging of the words according to the grammatical requirements of the target language. This step consists mainly in simply carrying out directions provided by the preceding step in which the grammatical units are recognized, but it also involves introducing words of the target language which have no equivalent in the source language, and dealing with the often troublesome problem of inflections. The question which I want to discuss is, How can these processes be programmed, and what operations are required of the computer in which they are performed ?

There are broadly two types of procedure available. There is the matching procedure, and the calculating procedure; or lexical and algorithmic methods, as they have been labelled in mechanical-translation circles. A pure matching procedure would deal with the three steps as follows. It would choose the translation of each word in the text by taking the word symbol of the input together with one or more symbols representing the information available about context, match the whole lot (presumably in series of stages) with headings in a suitable dictionary, and copy the word symbol entered in this dictionary against the given heading into the required position in the text. This, as outlined here, would require a very large 'dictionary', i.e. one with an enormous number of headings, though the information entered under each would be limited to a single word-symbol: it might nevertheless be practicable. As to the second step, of recognizing word groups, Mr. R. H. Richens in Cambridge has already worked out a practical matching procedure for doing this. His method consists in attaching to each word symbol of the input an indication of its grammatical function, in the form of a number indicating whether it is a verb, a noun, or whatever it may be, and listing in a word-sequence dictionary the possible sequences of these function signs together with their equivalents in the target language. The word-sequence dictionary is of quite manageable size, and its practical operation would present no insuperable difficulties. This procedure, of course, could be made to cover the second and third steps at the same time, since there is no reason why grammatical words of the target language, like 'the', 'is', 'of', and so on, could not be supplied by the word-sequence dictionary itself. However, matching procedures have certain intrinsic disadvantages. For one thing, they are slow, especially where large dictionaries have to be scanned, and the signs are that the commercial success of mechanical translation in its first stages at least will depend more on speed than on any other single factor. Another difficulty is that most commercially available computers are not very well adapted for the kind of matching we need, in which long digit sequences have to be compared, and in many of them matching programmes take up a great deal of storage space, and the advantages of being able to do mechanical translation on commercial machines, in the intervals between designing power stations and working out wages, are very great indeed, as compared with using machines designed specially for the job and used for little else.

We in Cambridge have therefore been particularly interested in developing algorithmic methods. I shall not describe all of these, for want of time and to avoid going too much into linguistic technicalities. But I would like to deal in a little more detail with our latest type of programme, in which both matching and calculating procedures are combined, because it seems to us to offer the best hope for a speedy solution of the mechanical-translation problem along the line of using so far as possible ordinary commercial computers. The basic principle on which our method rests is that it is possible to find a limited number of types of word group out of which all sentences and for that matter all extended texts are built up, irrespective of what language one is using. This useful result is based on a mathematical theory of language structure which is essentially an application of classical lattice-theory to communication by uniseriate signals between which relations other than their serial order can be significant. The root idea of using lattice theory for this came from M. Masterman and E. W. Bastin, and has proved of very great value. It turns out that each word group can be identified with one and only one of a set of patterns, each of which is mathematically a lattice whose elements correspond to potential words and whose partial-ordering relation can be defined strictly in terms of structural linguistics. In any given language, only some of the potential words thus defined are actually spoken, but in all cases enough of them are present unambiguously to identify the pattern which they make; different languages differ as to which elements of a given pattern correspond to words and which are 'latent', i.e. left out. For instance, as between the French 'Il me semble que c'est vrai' and its English equivalent 'It seems to me that's true', the 'que' of the French is left out in English, whereas the 'to' of the English is left out in French; but in this case it is fairly obvious that both represent the same pattern. This is much less obvious as between, say, 'Je n'ai que trois' and 'I've only got three', because here there are two words on each side with no exact equivalents, 'n(e)' and 'que' in the French and '(ha)ve' and 'only' in the English. In fact, the pattern of these sentences is quite complicated, and the difference between the assignment of words to elements between the two languages is quite considerable.

As to the practical application of the theory, we have found that the number of these basic patterns is quite reasonable; we have not yet worked out the complete list, but it is certainly well under a hundred, and since not all the theoretically possible patterns are actually used in all languages, the number can be reduced further. In Chinese, for example, we can manage with about 30, and in Italian probably about 40. Each pattern can be represented in the computer storage in various ways, but for speedy recognition the best method is to assign a number to each element and to represent the presence of each element n in a given pattern by the presence in its numerical representation of the digit 2^n. Allowing for 32 different lattice positions and keeping three digits for special indications (such as for repeated elements) we can thus represent each pattern by a 35-digit binary number. Each of these 'lattice positions' corresponds in linguistic terms to a particular grammatical function, and the corresponding numbers are entered in the dictionary as part of the word symbol to indicate what grammatical functions each given word is capable of having. For instance, a word like the English 'maple' is at once identifiable as a noun which we label 5, while 'bring' is equally definitely a verb and will have the label 6, and 'the' is the definite article, which we denote by 9, and so on. Other words have variable function, such as 'table', which can at least be a noun (5) and a verb (6), needing at least two separate dictionary entries. In finding the word patterns present in a given text we simply take in the sequence of these lattice-position indicators as they occur and by matching at each stage with the inventory of word patterns we discover, at particular points in the sequence usually corresponding to full stops or other major punctuations, what patterns have appeared. This information is stored, and used together with the word symbols in the later steps of the procedure.

Thus far this is a pure matching method. However, there is one point at least where calculation is required, and that arises because we find that any of our word patterns can itself take the place of a single word in some larger pattern; the place it takes is determined by a function number like those attached to individual words, and this number can be calculated from the function numbers of its elements. To do this we find we do not need in all cases the whole pattern complete, so that we can discover the function number of a pattern, sometimes at any rate, before we have identified it. This saves a good many matching operations and speeds up the overall procedure. We could in fact apply algorithmic methods also to finding the patterns themselves, but it appears that in this case the inventory matching method is quicker. This calculation, however, employs one operation which not all commercial machines are equipped to do, and could be made quicker if others could be provided. These special operations are those of Boolean algebra. The Cambridge Edsac I has one Boolean operation which they call 'collation'; it consists in writing a 1 wherever both the two numbers being operated on have a 1 and a 0 where either of them has a zero. Denoting this operation by O, we have for example O(10101, 11011) = 10001. The operation we mostly need happens to be this one, but it would help to have also I, which puts a 1 wherever either operand has a 1 and a 0 only where both have a 0, and X which puts a 1 wherever both operands have the same figure and a 0 wherever they differ. Thus I(10101, 11011) = 11111, and X(I0101, 11010) = 10000. These operations are electronically extremely simple to effect.

Actually, in the case mentioned, the function number belonging to any pattern is simply the O-resultant of all the function-numbers of its elements, *modulo* 16. In many cases we find that the translation of a word is affected if it occurs in a pattern having a function number different from its 'own' number. Thus the English word 'table' is a noun when and only when it is in a group whose O-resultant is 5 (mod 16), but in a group whose O-resultant is 6 or 7 (mod 16) it is a verb, and in most languages will then be quite differently translated.

The next step in the translation procedure is to find what word order the target language requires. This is in all cases determined primarily by the word pattern. Usually there is more than one word order possible in rendering a given word pattern, but if we are not concerned with literary style we can safely choose one of them in each case as an invariable standard. We can then regard each word-pattern discovered in the text as a direction to place the constituent words or word groups in a particular order and to add specified additional words not present in the original. Each such direction corresponds to a fairly short subroutine, and we can find each one by calculating an appropriate address number from the given pattern symbol. We have not yet discovered how much space in the computer store these subroutines will occupy, since we have not till recently had a full list of patterns for any language. But for Italian-to-English, on which we are now working, it appears that as many as 300 orders may be required.

This brings up the question of the size of storage necessary to work mechanical-translation programmes. As regards the dictionaries, these may have to be rather large, but can be regarded as belonging to the slow-access store. Probably the most convenient form in which to use this information would be on a magnetic drum as used in the Manchester computer; it is not yet clear whether magnetic tape, as used in the Cambridge Edsac, would give sufficient overall speed to make a commercially practicable process. Certainly the necessary equipment will not be available on any ordinary commercial machine, and to do translation one would have to bring one's own dictionary and in some way connect it up. This could in principle be done with the Elliott machines, for example, but with many less versatile models might present difficulties. On quick-access storage capacity any mechanical translation programme, if it is to be good enough to have any commercial prospects at all, will make heavy demands. It seems certain that more than 512 orders will always be needed, though there is a fair chance that 1 024 will be sufficient. One valuable feature of the type of approach I have been describing, based on an adequate mathematical theory of language, is that it makes possible the construction of translating programmes on an interchangeable-component basis. Thus, it will in general be possible to form a programme for translating from a language A into a language B by the putting together of three units, one common to all cases, one peculiar to A-as-source, and one peculiar to B-as-target. The two last may be again made up of components; thus, in French and Italian it is necessary to take account in general of whether an adjective follows or precedes its noun, and the operations connected with this form a single subroutine, which though needed for these languages could be simply left out in English or German programmes. The important and useful thing, however, is that the greater part of the programme will be actually common to all pairs of languages. Apart from the question of storage capacity and the faculty of making use of accessory storage representing the necessary dictionaries, there remains the problem of translating the actual programmes into the various order codes of particular commercial machines. If many different codes are to be taken into account, this can be quite laborious, especially where a given machine which it is desired to use does not have one or other of the operations which we need in its repertory. For example, our Boolean operations can be done on a machine equipped only for addition and subtraction, but each one needs a quite elaborate subroutine, and it is more than doubtful whether the type of programme I have been describing would be the best to use on such a computer. It does not therefore seem that there is any prospect of being able to do mechanical translation on just any machine that comes to hand. It remains to be seen whether ordinary commercial requirements will be such as to leave room in the market for the sort of computer which could be used for this purpose. We need a faculty of using additional storage up to very large total capacities, a quick-access store of at least 1024 'words', and an order system including at least one Boolean operation. We do not need much else: even addition, though used in our programmes, could be dispensed with, and multiplication and division are never needed at all. A large supply of counting registers is an advantage, but one is enough (our present programmes are designed for Edsac I, which only has one); also an advantage is a special section of store devoted to rearrangement of the contents of different locations.

Obviously, a complete machine with components serving as readers, dictionaries, computers and output writers could do the job of translation more efficiently and in the *long* run more cheaply than one designed for something else and pressed into service. I have myself no doubt that in due course such machines will be manufactured. But in the immediate future the question of overheads and capital costs will preclude this solution. Mechanical translation will at first be a spare-time occupation for the larger commercial computers, and what I would like to ask those with experience of the commercial computer field is, would any manufacturer now in the market find it worth while to keep mechanical translation in mind as a possible extra attraction to purchasers, if this could be made possible by a reasonable modification of existing designs ? I would count it as 'reasonable' to add the Boolean O-operation to the order code, for example, but clearly it would not be 'reasonable' to adapt a small machine to cope with a very large dictionary store if not provided for in the original design. It is along such lines that in my opinion mechanical translation is most likely to develop.