AN ALGEBRAIC THESAURUS

A.F. Parker-Rhodes

Introduction

It is clear that one difficulty about attempting to use the thesaurus principle in mechanical translation will be that of storage of and access to the information representing the thesaurus itself. While not necessarily insoluble, it is clear that it would be an advantage if some at least of the storage capacity of our computer could be saved by means of some device to reduce the amount of thesaurus information needed. Moreover the use of such a store will inevitably involve a good deal of looking-up-work, which is a relatively slow and therefore inefficient way of using the resources of an electronic computer of whatever size. The more this can be replaced by computing work in the strict sense, the better value we are getting for the capital outlay that a computer represents.

It is the purpose of this paper to suggest that this problem may be completely soluble, in the sense that, with sufficiently adequate theoretical work and painstaking compilation of information, we need carry nothing at all in the machine storage corresponding to any actual thesaurus, and do the whole of the work by computation. In this way a complete translation programme would, if the device were successful, consist of five stages, (i) input of text source, (ii) a one-to-one matching routine to find code equivalents for the "words" of the input text, (iii) a sequence of arithmetical operations on these code-numbers, (iv) a second one-to-one matching routine, finding words of the target language corresponding to code-numbers which will have been computed in stage (iii), and (v) output of printed translation. Of these stages, (iii) will be probably by far the quickest, and (ii) and (iv) will be rate-controlling, though as one-to-one matching routines they present no technical problems other than engineering ones.

At first sight it would seem that this is an impossible objective, and in terms of complete practical realization it probably is so; but it is far less utopian than might be thought, since in certain definite and encouraging senses the mathematical problems involved are demonstrably "soluble". The gain from a successful solution of the mathematical difficulties is so evident, that it is worth quite a lot of theoretical work to get out an answer; but of course the practical question, can such a method be programmed at reasonable cost, will not be answered except by practical experiment.

The Basic Principle

In order to find the correct translation for a given word, or morpheme, or word-group, or sentence, appearing in a given written text, we have the following data to go on. First, the word or group itself; in a good many cases, technical terms for instance, this may be by itself sufficient to give a unique and satisfactory translation, but this will not be the case always, and especially not so with the commonest words of any language, which by frequent usage tend to acquire a large retinue of metaphorical and altered meanings, and also tend to have a higher frequency of homonyms than the less frequent words. Second, we have the words with which it is associated; what we mean by "associated" I will discuss further below. Third, the subject or general background of the text. A fourth term, the general cultural pre-suppositions of the language of the text, ought perhaps to be added, in which we should include such things as literary quotations and allusions, assumptions in regard to social custom, and the like. This category of data will only really be of importance in literary texts, which are still beyond the immediate horizon of M.T., but it need not be dismissed out of hand as irrelevant even in the dullest species of writing. However, I shall not take account of it in this paper.

These three terms I shall here call the contexend, the minor context, and the major context respectively, ignoring the cultural context. The aim is that given a particular contexend, minor context and major context, we should be able to find either (a) a unique word or phrase of the target language as a suitable rendering of the contexend, or (b) a set of such renderings between which we can choose at random (or subject to an ancillary stylistic programme), or (c) the result that the contexend is untranslatable. In the latter case translation would have to be attempted with a longer segment of discourse including the contexend with which we have at first failed. This last will happen quite often, whenever in fact we have to do with an idiomatic expression, or one whose grammatical structure has to be recast before it can be expressed in the target language.

In the purely lexical type of thesaurus the desired result is achieved by looking up the relevant words or other units in lists prepared beforehand. A purely algorithmic thesaurus would replace all such lists by rules of calculation. The question is, what would such rules be like?

They would evidently have to have certain very definite mathematical properties, if they were to work. First, the compiling of the lists in a lexical thesaurus is obviously not a mere waste of time, and so there would have to be some mathematical analogue of this process. Call it the operation A. Then we can envisage the process of going through some huge corpus of texts in the source language noting down the minor context, say, of every word on its every occurrence, so as to discover all possible contexts of each word, and this will be the result of our operation A. Next, we shall have to find the particular minor context for each word or group in the given source text by another operation, which we may call C. Then, when a given word has been read in the input text and the minor context found, we must perform an operation B on these two so as to get a result which will have to be a unique releaser of one or other of the three admissible answers listed above: one target equivalent, a set of equally acceptable equivalents, or no equivalent at all. The result we have to avoid is getting two or more target equivalents without any reasonable confidence that they are equally correct. Statistical probabilities just aren't what we want: the human translator does not deal in probabilities, except in very obscure texts, and to rely on being probably right is far too like the practice of the slovenly schoolboy to be an acceptable aim in M.T,

The Mathematical Requirements

The first definite property the operations A, B, C must have is as follows. Let us denote the code-equivalent delivered by the input dictionary for a given contexend by a (because it will be the result of an operation A performed by human operators before we start); and the symbol standing for the minor context, calculated by an operation C, by c. Then the code equivalent of the contexend in its context we desire to be B(a, c). This is the symbol that stands, in our code, for what "a" in this context "means". If we had all the possible meanings of this same word or group in all possible contexts of the source language and combined them together the result would be precisely what we hope to achieve by the operation A. Suppose these several constituent "meanings" of the word be represented as $u_1, u_2, \ldots u_m$. Then what we have said is that

$$A(u_1,u_2,\ldots u_m) = a \qquad (1)$$

and at the same time

$$B(a, c) = u_i \qquad (2)$$

where u_i stands for some one of the "meanings" u. Therefore

$$B\ (\ A(u_1,u_2, \ldots u_m), c\ ) = u_i \qquad (3)$$

is a perfectly general relation that A and B will have to obey, whatever mathematical form they turn out to have. It is in fact quite a restrictive one. It requires for example that B must be distributive over A.

It also requires, since the assignment of the symbols u to the meanings is obviously arbitrary, that A is both commutative and associative.

There are a number of other relations like (3) which we can discover as necessary if the procedure is to work. Our problem is then the classical problem of the applied mathematician: given a certain set of symbolic expressions, find a self-consistent non-trivial mathematic in which they are all true. The expressions (l) to (3) are not strictly mathematical expressions, since their truth is defined to be independent of the interpretation of the symbols; they are syntactic expressions. If we add the additional condition that all the operand symbols in the expressions should be capable of being any members of a set which forms a group under all three operations A, B, C, it is already evident that no algebra based on the rational numbers will satisfy the conditions. Is there any algebra that will?

A.Pilot Solution

Most probably there are an infinity of solutions. There is one at least which we don't have to look far for. The kind of mathematics which is easiest to do on an electronic computer is Boolean algebra, and a solution exists (in fact, a whole set of solutions) in terms of this. It can in fact be shown that if p and q are any two elements of such an algebra, and the operand symbols of (1) to (3) are taken to be elements of the same algebra, then a solution of the problem is given by putting

$$A = \Phi_p; \quad B = \Phi_q; \quad C = \Phi_{-p}. \qquad (4)$$

where -p stands for the complement of p and where the result of an operation Φ_n is defined by putting, in every binary place where n has a 0, a 0 unless both operands have a 1 in that place, and in every binary place where n has a 1 putting a 1 unless both operands have a 0. This reduces to the operation of class-addition when n is the I-element of the algebra and to the operation of class-multiplication when n is the O-element. Although both p and q are arbitrary this solution is in other respects unique within the scope of Boolean algebra.

It is however not sufficient to find a solution for the formal problem, if the solution found is either trivial or impracticable. The solution given above is in one sense both trivial and impracticable. It would indeed be very surprising if the answer to so difficult a problem as we have seemed to set ourselves were so exceedingly simple. Briefly it is trivial because, for one thing, it takes no account of the major or background context which is the only kind of context most M.T. workers had thought of till quite recently. Why it is impracticable will appear when I have worked an example in it.

So I can't say we've got the answer all ready for you to take home. But I hope before the end to show in general terms how both the triviality and the impracticability of the pilot, solution may be overcome, by inventing a new sort of mathematic more suitable for our needs than simple Boolean algebra.

An Example Worked Out

To show how far we can get with this pilot solution, I shall take the case of the phrase which came least well out of Miss Masterman' s analysis, and see whether we can tidy it up this way. This is the Italian "per le piante di fibra", which came out, at the output-no.2 stage, as "for the plants of fibres". This rather poor result is not due to any incapacity of our programme to effect a reordering of words, which it does very well (in English a prepositional clause qualifying a noun always follows the latter, and that is what, if anything, "of fibres" is.). Nor is it due to the words "plant" and "fibre" being wrong renderings of their Italian originals, which they are not. The trouble is simply that we haven't found a way of demonstrating, in any way which does not bring in ad hoc assumptions and so falls short of generality, that the Italian prepositional clause has got to be rendered in English by an attributive noun. I will now try to show how

the method suggested as a pilot solution of the algorithmic thesaurus problem can solve this difficulty.

To make the demonstration practicable it is necessary to restrict rather severely the number of "ideas" which we consider relevant. The following meagre selection is all I shall have space for. Each "idea" is to be represented by one binary place in the numerical symbol for each element, in which place there will be written a 1 if the corresponding idea is present in the given element or a 0 if it is absent. I shall take as my algebra the Boolean algebra which has 32,768 elements, each of which can be represented as a binary numeral of 15 digits, and I shall put the q of the pilot solution equal to "15" (i.e. to the element 00000,00000,01111), while for p and -p 1 shall take the I- and O-elements respectively ("32,767" and "0"). This means that the result of an operation B(x, y) will have in each of the first eleven digits a 0 unless both x and y have a 1, and in each of the last four digits a 1, unless both x and y have a 0. A will operate like the last five digits throughout, and C will operate like the first ten digits throughout,

First we must perform the operation A to get a code-number for each of the five words of the text phrase, excepting "le" which we treat as a pure operator and deal with by the lattice programme entirely. It will not be necessary to go through the whole of Italian literature to do this, because I think you will agree without much argument about the classification of each of the words in respect of the limited repertory of ideas here examined. These categories are on the lines of those suggested by R.H. Richens, though their development here is somewhat different.

Personal behaviour	1	Places	6	Things to think about	11
Physics	2	Persons	7	The idea not "di"	12
Botany	3	Agriculture	8	The idea not "per"	13
Object (not action)	4	Social affairs	9	The idea not "fibr-"	14
Morals	5	Abstr.quality	10	The idea not "piant-"	15

It will be noticed at once that the last four digits, where q has a 1, are of a different kind of content from the first 11 where q has a 0.

Now our A is to be the ordinary Boolean I-operation, commonly called "cup". It puts a 1 wherever either operand has one. So it puts a 1 against each word to be coded in the place corresponding to each "idea" which in any context it can represent or contain. For example the word "piante" in Italian can be a part of a verb meaning "to weep", so we have to give it a 1 in the personal behaviour column. It never means anything to do with physics (or so I assume here), so in the second digit it has a 0; and so on. Among the last four digits the only one it never has anything in common with is its own negation, so all but the last have 1's. In this way we make the code-number for "piant-" to be 10110,00110,11110, I need not go through all the words in detail: the results are as follows:

"per"	:	11111,11111,11011
"piant"	:	10110,00110,11110
"di"	:	11111,11111,10111
"fibr-"	:	11111,00101,11101

It will be noticed that the two prepositions have a 1 in every place except their own negation; that is because prepositions (these prepositions anyway) can occur in all contexts. Although I said before that this method doesn't take any account of background context, it is easy enough to bring it in for purposes of illustration as though it were the code-number of an extra imaginary "word" present in each of the lattice groups we have to analyse. The code-number of this word is clearly fixed by what you have already seen of the passage from which this phrase comes. It is the introduction to a paper about breeding tobacco without axillary buds; it thus has to do with botany, concrete objects, agriculture, abstract qualities, and things to

think about, but has no reference to personal behaviour, physics, morals, places, persons, social affairs. The last four digits are all 0's since no major context may include the negation of any idea occurring in it. Thus we add to our repertory the background context element:

00110,00-101,00000

The next step is to form from these the minor contexts by means of the operation C, which is here simply the O-operation or "cap". It will put a 0 wherever either operand has a 0, and a 1 only in those positions where both have a 1. The minor context has to be formed separately for each lattice group. In this case the only lattices involved are simple chains so that no complications arise from this source, as they do with more complicated lattice forms. The first group entered upon contains two manifest elements, "per" and "le piante di fibra"; we can't form the minor context of this till we have that of its second element. This has three elements, the nugatory "le", "piante", and "di fibra"; we still have to analyse the last one into its two manifest elements "di" and "fibra". The minor context for this lattice is therefore C("di", "fibra") which is readily found from the above list to be

C(11111,11111,10111, 11111,00101,11101) = 11111,00101,10101

in like manner we find the minor contexts for the other two lattices, with the results:

"di fibra"	:	11111,00101,10101
"le piante di fibra"	:	10110,00100/10100
"per le piante di fibra"	:	10110,00100,10000

We are now in a position to perform the last operation, which ought to deliver the code-numbers which shall represent in our output dictionary the English words required for a proper translation. Before we do this however, in case you accuse me of cheating, it will be as well to decide on the sort of code numbers which will represent a selection of plausible English words.

Some Output Code-Numbers

Let us start with the word "di". No doubt "of" is the English form which fits it most often, but in other contexts it can be rendered as "from", "belonging to", "part of", and many others; of course, by using the thesaurus method we are never limited to a particular pre-arranged set of "readings", but on suitable occasions any of these renderings could be replaced by another equivalent or near-equivalent form. I should make it clear that this analysis of the preposition "di" is independent of the type of analysis which Dr Halliday has worked out in our unit, and which, when it has been reduced to the proper symbolic forms, will supersede this rather crude method.

With the limited repertory of ideas I am working with in this example, some of the possible renderings of "di" in English could be defined by the following code-numbers:

"concerning"	:	00000,00001,10111
"part of"	:	01110,00100,10111
"from" (referring to origin)	:	00000,10001,10111
"belonging to" (ownership)	:	00000,01000,10111
"qualified as" (adjectival)	:	11111,10111,10111

These numerals are derived by performing the A operation on the whole set of contexts in which the given rendering could be appropriate: for example, the figure for “part of” has a 1 in each place dedicated to ideas proper to things which have parts.

In our test phrase, "di" occurs in the lattice group "di fibra", whose minor context we have already found to be 11111,00101,10101. Taking this together with the code-number of the word "di" itself, which is 11111,11111,10111, and the major context 00110,00101,00000, we perform the B operation and get

B ("di", minor context, major context) = 00110,00101,10111

This element does not equal exactly any of the specimen renderings of "di" given above; in general we can't expect ever to get an exact match, at least with so polysemantic a word as a preposition, but we can easily infer the rule by which to get from a particular code-number for which no equivalent appears in the output dictionary to the nearest acceptable equivalent. A more general term can always if need be replace a less general one whose semantic field it includes, but the contrary does not hold. Therefore, since the code-numbers are formed by A operations, a given code-number can be replaced by another having 1' s in at least those places where it has them itself; that is to say by an element which is greater than it in the lattice sense. Now in our example the list of equivalents for "di" given above represents part of the output-dictionary concerned with this word. If it were the whole of it, evidently the only entry acceptable as a substitute for the one calculated would be the last one, in which "di" is rendered as an adjective-forming function.

The word "fibra" is of course easier to manage. Its possible renderings may for the sake of illustration be reduced to three:

"fibre, sinew"	01110,00100,11101
"strength, muscle"	11000,01001,11101
"courage"	10001,01001,11101

The code numbers can easily be checked by reference to the table of values of the digits; their A-resultant is equal to the code number already given for "fibra". Within the "di fibra" lattice-group, we get no further light on the meaning of this word, since its only companion is the wholly uninformative "di"; but on passing to the next higher lattice group, "piante di fibra", we have the minor context 10110,00100,10100, which with the major context 00110,00101,00000 gives under a B operation:

B ("di fibra", maj. context, min. context) = 00110,00100,11101

and this is less, lattice-wise, than only one of the entries in the skeleton output-dictionary given above, namely the one for "fibre". The alternative "sinew" could easily be excluded by reference to the non-zoological character of the context. This information relates not to the particular word "fibra" but the lattice-group "di fibra"; since we have already discovered that "di" here has the force of an adjective making particle, the translation required for the whole group will be "fibre as an adjective": in pidgin-form, "fibre-y". A further point can be added in the output-dictionary at this stage, namely the information that this kind of adjective renders the noun-group to which it belongs incapable of carrying the definite article in English. This information forms part of the LPI in our lattice programme, and if the thesaurus operations are carried out immediately before each lattice group is contracted such modifications of LEE can be taken account of in constructing the lattices, and in this way the unwanted "the", other-wise foisted on us by the "le" of the source text, can be got rid of.

By precisely similar means we find out at the next lattice stage that "piante" in this context has to be translated "plants", and has nothing to do with weeping, pitching tents, incriminating accomplices, or posting sentries, to mention only a few of the surprisingly many possibilities which the dictionary reveals. Finally, "per" comes out to mean "with" because we have incorporated into the background context the information that we are dealing with things to think about (though with a more realistic repertory of ideas of course this would not be at all sufficient to arrive at the conclusion).

Finally then we arrive at the translation "per le piante di fibra" = "with fibre-y plants". The transposition of word order is produced automatically by the lattice programme as soon as we decide that "di" is here equivalent to an adjectival suffix. The result is in effect correct, for we suppose that in actual practice the pidgin-stage of the output as given here (representing the stage when everything is represented by output code numbers) will not be formulated, but its elements will be put across one-to-one in the output dictionary, so that fibre-y will be replaced by the correct form "fibre".

Why the Pilot Solution Will not Do

By this time no doubt you will be convinced that the apparent very mild success produced in the above calculations is all due to the highly artificial selection of ideas on which it is based, and that to work the method properly we should need vastly more digits and then, even if it could be got into a computer, the method would become too cumbersome to use even if it would work at all. So indeed it would.

From the example just given it is rather obvious that to generalize the method adequately we should need so many digits that each code-number would need something of the order of a thousand bits, and although the engineers are making rapid progress in the design of storage systems this is still a bit above what we can reasonably manage. And in any case the cost of storage capacity is roughly proportional to the total number of bits, so that there is a heavy premium on economy of space. And of course the people who have to compile the dictionaries, input and output, in which the essential thesaurus information on this method lies concealed, will not like having to deal with binary numbers of such length.

Not only that, but some of you may have detected a cheat in the way the major context was dragged in. Very little experience in working examples will show that this way of doing it just won't work, except when as here it happens to more or less by accident.

Therefore the "solution" proposed to the problem of an algorithmic thesaurus turns out to be no solution worth having. But it is not all done by cheating, and it is not a complete waste of time working an example like the one just -gone through, because in revealing the inadequacies in the pilot solution it points out the way to a better one. Two deficiencies have to be made good. First, we have got to take into account the major context as a separate and independent element in the available data, as it in fact is, and not as an imaginary word added to each lattice group to supplement the minor context. Very little consideration is needed to see that the logical effects of these two types of context are or should be dissimilar, whereas the way of doing it in our example makes them symmetrically related which is obviously wrong. Secondly, we have got to find a way of compressing the information into fewer digits than are needed in the simple method outlined. This means abandoning Boolean algebra, which is a pity, as it is the simplest possible kind of mathematics from the electronic point of view and the most elegant from the mathematicians' standpoint. A new algebra will have to be invented, related to Boolean as nearly as may be possible, but non-trivially different. The kind of algebra we want will be based on a compressed representation of our code-numbers, analogous to one of the nabla (∇) operations introduced for word coding by the Soviet workers. Actually none of the nabla operations mentioned in Acad. Panov's paper will do here, but they show the way.

The algebra which I am now working on, and currently hope will prove
workable, may be regarded as the derivative we get from Boolean algebra when we replace the binary numerals by which we commonly represent the elements of Boolean algebra by numerals to some higher base, but go on performing Boolean-type operations on them; operations, that is, in which the carry-over principle employed in rational addition and subtraction is not allowed. This algebra is rather a lot more complicated than Boolean, and it is possible that its arithmetic may prove to be very irksome, though scarcely so to a well-designed computer. This is the more so as

in order to bring in both kinds of context adequately we shall need to use ternary operations which are not reducible to binary ones. There are over a hundred families of ternary operations (excluding degenerate ones which are equivalent to binary ones) in ordinary Boolean algebra, so that finding an adequate solution to our problem in some appropriate meta-Boolean algebra will be fairly heavy going. But the chances of success are pretty good, and, as I said at the beginning, if it should be possible to reduce the whole of the thesaurus stage of the translation programme to algorithmic terms, mechanical translation would be revolutionized. It would in fact become a commercial possibility forthwith, which is still far from being the case except perhaps for very limited purposes.